NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTOMATED SUPPORT FOR RAPID COORDINATION OF JOINT UUV OPERATION

by

Seneca R. Johns

March 2015

Thesis Advisor:	Dennis Volpano
Second Reader:	Kevin B. Smith

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704–0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2015	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE AUTOMATED SUPPORT FOR RAPID COORDINATION OF JOINT UUV OPERATION			**5. FUNDING NUMBERS**
6. AUTHOR(S) Seneca R. Johns			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Recently, marine services company Phoenix International headed the search efforts for Malaysian Airlines flight 370 using its Bluefin-21 autonomous unmanned underwater vehicle (UUV). In total, it conducted 270 hours of in-water time and covered approximately 250 square miles of ocean floor. Deploying multiple UUVs simultaneously would have increased the coverage area substantially within the same time period. Ideally, a coalition of countries would be able to jointly deploy their autonomous UUVs with little or no advance preparation since search time is limited. Such a task is beyond today's capabilities. Multiple UUV coordination today relies heavily on acoustic communications, advance preparation and manual guidance. This thesis explores the application of static analysis to allow multiple UUVs to be deployed simultaneously with little advance preparation and no acoustic communications.

14. SUBJECT TERMS UUV, AUV, acoustic communications, finite-state machine, Moore automata, dead-reckoning, static plan, runtime plan, rapid deployment, GPS			**15. NUMBER OF PAGES** 73
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AUTOMATED SUPPORT FOR RAPID COORDINATION OF JOINT UUV OPERATION

Seneca R. Johns
Lieutenant, United States Navy
B.S., Thomas Edison State College, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2015

Author: Seneca R. Johns

Approved by: Dennis Volpano
Thesis Advisor

Kevin B. Smith
Second Reader

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Recently, marine services company Phoenix International headed the search efforts for Malaysian Airlines flight 370 using its Bluefin-21 autonomous unmanned underwater vehicle (UUV). In total, it conducted 270 hours of in-water time and covered approximately 250 square miles of ocean floor. Deploying multiple UUVs simultaneously would have increased the coverage area substantially within the same time period. Ideally, a coalition of countries would be able to jointly deploy their autonomous UUVs with little or no advance preparation since search time is limited. Such a task is beyond today's capabilities. Multiple UUV coordination today relies heavily on acoustic communications, advance preparation and manual guidance. This thesis explores the application of static analysis to allow multiple UUVs to be deployed simultaneously with little advance preparation and no acoustic communications.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AOSN	autonomous oceanographic sampling network
ASW	antisubmarine warfare
AUV	autonomous underwater vehicle
bps	bits per second
CoDA	cooperative distributed AOSN
CSG	carrier strike group
CTF	combined task force
DON	Department of the Navy
GPS	Global Positioning System
ISR	intelligence, surveillance and reconnaissance
MAUV	multiple AUVs
Mbps	megabits per second
MOOS-IvP	mission oriented operating suite interval programming
ROE	rules of engagement
RF	radio frequency
UUV	unmanned underwater vehicle
VBAP	virtual bodies and artificial potentials

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I acknowledge, with gratitude, the patience and tireless efforts of my advisor, Professor Dennis Volpano, for keeping me on track and assisting me with the completion of my thesis. His involvement was crucial to my work in order to convey properly the unique language (pun intended) that comes with the application of automata. No one has more experience and could have done it better.

Most importantly, I would like to thank my wife Aimee and daughter Ryan for their constant love and support. To Ryan: You have my heart in the palm of your hands. Know that I will always be there for you, an advisor in life. To Aimee: I cannot express in words how important your support has been while here at NPS. Thank you for being my best friend. I owe you everything.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Recently, marine services company Phoenix International headed the search efforts for Malaysian Airlines Flight 370 in the Indian Ocean with its Bluefin-21 autonomous unmanned underwater vehicle (UUV). Traditional submarine support for this sort of mission would be helpful, but is not economical. The Bluefin-21 UUV can operate at far less cost than a submarine and cover just as much area. According to [1], it covered approximately 250 square miles during the search. Its primary drawback, however, is the time it takes to cover such a large area. If the Navy could deploy multiple UUVs simultaneously, then the coverage area would increase substantially within the same time frame. Moreover, we would like UUVs to operate autonomously, since navigating them manually usually requires additional resources, like a vessel at sea. Though autonomous, UUVs must operate as a team [2].

The basic problem this thesis addresses is how to coordinate multiple UUVs in the context of a rapid deployment in which the UUVs come from different vendors—and are thus unlikely to communicate with each other—and there is little if any time for UUV preparation or planning before deployment. These circumstances could arise, for instance, if a coalition of countries rapidly marshalled their UUV resources to search for a downed airliner. Our approach to solving the problem is novel. It is based on analyzing by computer static descriptions of the *executable* navigation plans of UUVs and deciding in advance of deployment whether any plans conflict. We say two UUVs conflict if there is risk of collision or if one can pass the other in a way that interferes with operation, like generating too much noise.

A. NEAR-TERM APPLICATIONS OF UUVS

UUVs are attractive in that they can eliminate the threat to humans working on or under water. An example would be mine hunting, an operation in which deploying several UUVs would eliminate risk to human life and perform the tasks as required. Aside from search missions, persistent presence on station provides accurate and consistent data collection in all ocean environments, supporting real-time operations as well as intelligence preparation of the battle space for expected operations. The Navy UUV Master Plan [2] identifies several other key intended uses for UUVs and highlights their importance and integration into the fleet as a valuable asset. The image in Figure 1 depicts the Navy's vision for UUV integration into the fleet.

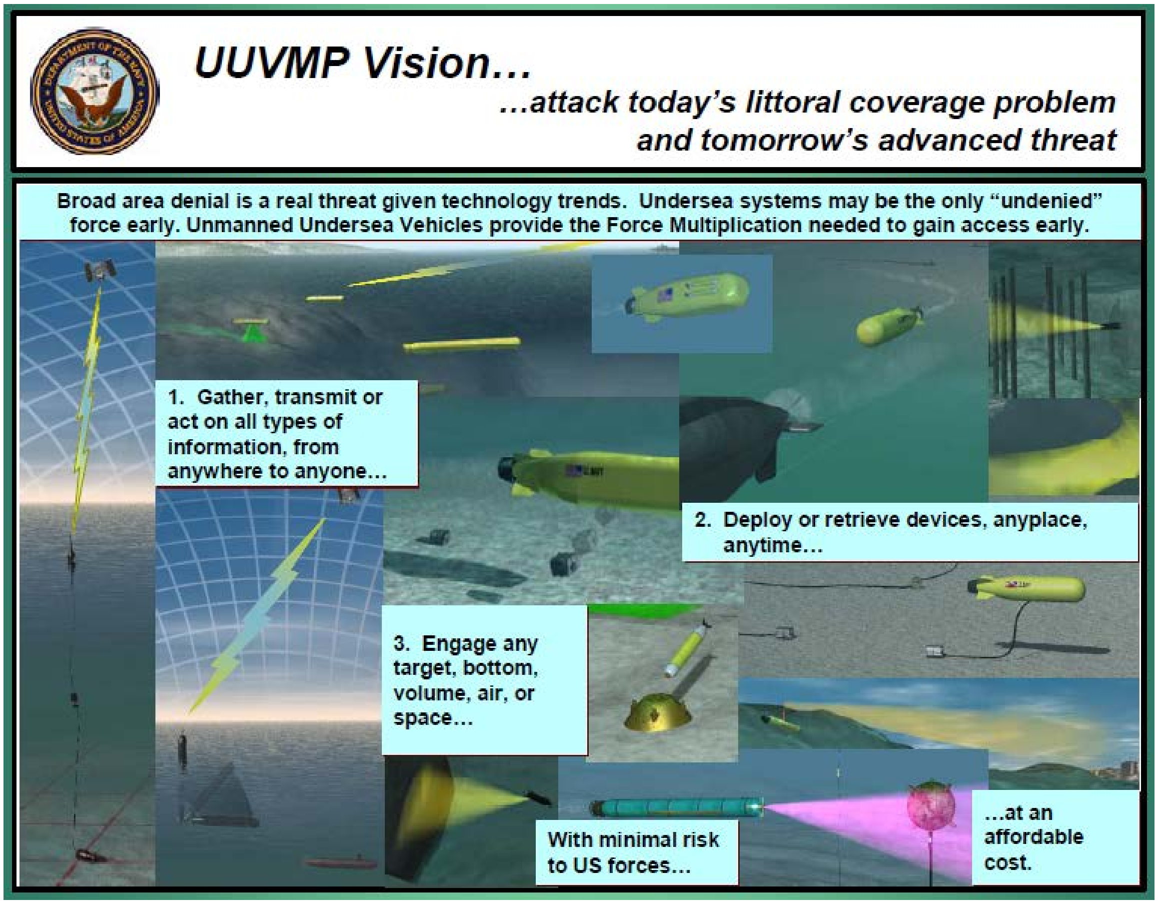

Figure 1. Expanding the Role of UUVs to Meet the Navy's Mission, from [2]

Of great interest is the ability for UUVs to conduct intelligence, surveillance and reconnaissance (ISR) missions, provided the communications piece is realized. Their ability to remain on station for extended periods—on the order of months—is extremely valuable in terms of data collection and would make them a valuable asset to the Navy's mission.

Communication nodes and navigation network nodes support the critical communications link between subsurface, surface, land and air assets. UUVs could potentially traverse areas unnavigable by conventional methods, allowing for undersea networking and close ashore surveillance. That data could then be relayed back to the host ship. Although it is not a primary mission of UUVs, their ability to act as communications relay nodes could eliminate limitations in over-the-horizon communications. Future uses include large networks spanning all dimensions of warfare (subsurface, surface, land and air) providing complete battlespace awareness that exceeds current capabilities.

UUVs could provide advanced sanitization of waterways intended for safe operation and transit of critical assets such as the carrier strike group (CSG) (e.g., entrance to the Strait of Hormuz). Warships inherently operate with additional risk and, in the case of a carrier, present a large target for adversaries. Warships transiting a choke point are at much higher risk than in open water. Ideally, the theater combined task force (CTF) would task submarine support to patrol the waterways and the choke point entrance prior to the CSG transit; however, submarines are not always available for tasking by the theater CTF, as national tasking generally takes precedence. If they happen to be available, the window is generally very small and the collected data becomes old and unusable quickly. A more persistent presence by UUVs could mitigate this problem entirely, providing data to the CSG before, during and after the transit.

UUVs utilized as mine countermeasures eliminate the risk to human life and enable the Navy to continue power projection ashore, in contested waters, and against increasingly belligerent adversaries. Mines present a large risk to manned vessels and are of significant concern for many reasons, but, strategically speaking, can deny access to enemy waters. The Navy operates on power projection and presence throughout the world in all waters navigable. If an adversary were to deny access to their waters by way of mining, this would present a serious challenge to naval forces. UUV operations in mine countermeasure warfare are probably one of the most effective uses of these vehicles in their current form; however, this does not necessarily scale well, since there is no effective way to deploy multiple UUVs operating in relatively close proximity without risk of collision or interference.

UUV deployment in an antisubmarine warfare (ASW) capacity eliminates the risk to human life, has the potential to collect critical operating data on the adversary's forces and their movement, and can deliver the data to friendly forces in a timely manner. The potential for offensive payload delivery in the ASW realm exists as well, although it is a difficult concept to implement due to restrictions in the Navy's rules of engagement (ROE). Underwater gliders based on buoyancy engines to provide propulsion are relatively quiet in the water and difficult to detect by submarine sonar systems. Deployment of these types of UUVs in an ASW capacity would provide significant support to conventional forces operating in contested waters.

B. AUTONOMOUS JOINT UUV OPERATION

The applications of UUVs discussed so far could be carried out through remote manual control. In the long term, UUVs are expected to operate *autonomously*, meaning without manual control. Moreover, they are expected to be deployed *jointly* to form a team of autonomous vehicles working on a common task. These two added dimensions pave the way for more cost-effective ways to search for undersea threats and support civilian use cases like searching for downed aircraft. However, autonomous multi-UUV coordination is challenging for many reasons. We outline the major ones here.

1. Sound Propagation in Water

Communication is very constrained under water due to limits of sound propagation. Underwater communications are highly dependent on environment variables (e.g., temperature, depth, salinity and weather) and have a limited range. The image in Figure 2 depicts sound propagation in a typical underwater environment. Although there has been research in establishing underwater communications, as demonstrated in [3], [4], [5] and [6], these communications are severely limited by the propagation of sound in water.

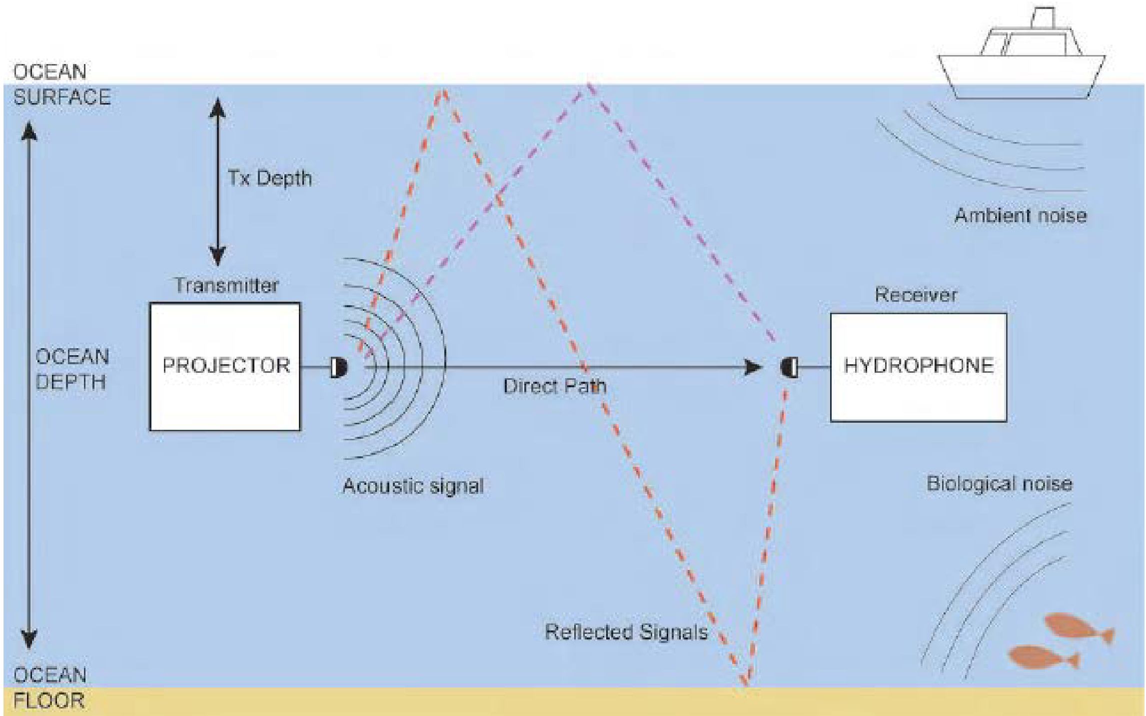

Figure 2. Sound Propagation in a Typical Underwater Environment, from [7]

Current acoustic modems, such as the Teledyne Benthos 900 series commonly used in UUV applications, are limited to a range of 2 to 6 km, depending on the environment. Underwater sound propagation is dependent on losses that are both range and frequency dependent, according to Burrowes et al. [7]. Constraints include high latency with acoustic signal propagation through water—roughly 200,000 times slower than that of signal propagation in air—and signal fade due to absorption and multipath. Higher frequencies would mitigate some of the ambient noise issues; however, this would affect range, as higher frequencies fade more quickly and require more power.

2. UUV Power Supply

Another inherent constraint of UUV operations is battery capacity. This requires the Navy to consider the type and number of sensors installed for a particular mission and its duration, since different sensors have different power requirements. Although higher frequencies would help mitigate the ambient noise, it requires more power to transmit and for the transmitter and receiver to be closer. The higher frequencies are also more vulnerable to absorption and attenuation, thus reducing propagation. This is not an issue if the mission duration is relatively short; regardless, power consumption should always be a consideration in UUV design.

3. Bandwidth Constraints

Another constraint of operating in water is limited bandwidth. Current rates among commercially available acoustic modems produce around 360 bits-per-second (bps), with an error rate highly dependent on the signal-to-noise ratio. For a real-world comparison, a typical 4G connected mobile device can observe speeds between 5 to 12 megabits-per-second (Mbps), which is roughly 20,000 to 30,000 times faster than the speeds capable in water. This limits the amount and type of data that UUVs can transmit between one another and requires complex signal processing and error correction after receiving the transmission.

4. Joint Navigation

Each UUV is programmed for a maneuver. Coordinating multiple UUVs for joint operation requires manually ensuring that each maneuver, as codified by an executable program for navigation, doesn't interfere with another UUV's programmed movement. This might be established by showing disjoint paths or steps that each UUV takes to avoid collisions, in the event that they can communicate during the operation. As this is a manual process, it does not scale to large deployments and there is no guarantee that all potential interference will be detected, since the process is very prone to human error. Programs are typically expressed in code like C++, which makes analysis difficult.

C. AUTOMATED SUPPORT FOR RAPID JOINT OPERATION

Our approach envisions each UUV operator constructing a plan for that UUV's underwater operation. A plan controls a UUV based on inputs such as GPS waypoints and course/speed definitions. It allows extended operation, including object avoidance capabilities. The plans are codified in a way that a computer can automatically detect any conflicts that exist before UUVs are deployed.

1. Sampling Rates

The key observation is that if multiple UUVs sample their locations at the same rate, then under predictable operating conditions, their plans can be compared for conflict in advance and if none is found, deployed according to their individual plans. Further, if a conflict is detected, our approach will indicate where and when in their combined operation it would occur. This gives operators insight into how the conflict might be resolved, for example, by changing the speed of one UUV.

The ideal use case for our approach is multi-UUV deployment by a coalition in which rapid deployment is needed (e.g., a downed airliner) with no advanced preparation. Plans for UUVs participating in the coalition must not conflict.

2. Limitations of the Approach

There is no real-time communications link between UUVs or to a mother ship. The goal is to deploy multiple UUVs jointly and guarantee a priori that a conflict does not exist. UUVs do not communicate once deployed. However, in the case of unpredictable operating environments, some communications may be advisable in order to handle those situations in which a UUV must recover from an event it did not account for in its plan.

This approach will not factor in depth when considering possible collision scenarios. While two UUVs can share the same space in terms of three dimensions, for the purposes of this thesis, only a two dimensional layout is considered. This limits all UUVs to operate at the same depth when using our approach. There's nothing inherently difficult in treating depth as well; however, for this thesis, only two dimensions were considered during calculations. Implementation of the third dimension is discussed in Chapter V.

This approach is based on statically analyzing navigation plans of UUVs. These plans are constructed according to the capabilities of a UUV operating at a certain speed, depth and initial direction under predictable operating conditions. Thus, any conclusions with regard to conflicts in the future based on them will only be as accurate as these parameters remain constant. Changes, for example, in UUV speed or direction induced by the environment during operation and not considered in a navigation plan make obsolete any type of static analysis done a priori.

Our approach assumes that every UUV when operating jointly operates at a constant velocity. A UUV may change its direction but not its speed. This is a tradeoff for being able to detect conflicts in advance while not requiring any communication between UUVs during their joint operation. An approach to allowing for planned changes in speed is discussed in Chapter V.

Minimum UUV navigation capabilities include, but are not limited to, a GPS transponder and a dead-reckoning ability. For this thesis, a generic, buoyancy-engine-based UUV was assumed for modeling and simulation. Accuracy in location is dependent on user-specified GPS-fix intervals. During each fix the glider will reorient itself and attempt to either maintain or regain intended track.

D. ROADMAP

The remainder of this thesis is organized as follows. The static analysis of executable navigation plans calls for representing these plans in a way that a computer can reason about automatically. That means plans cannot be expressed in languages like C++. Instead, plans are expressed as finite-state machines known as Moore automata. A definition of Moore automata is given in Chapter II. Combining Moore automata for the purpose of detecting conflicts between plans calls for a new form of product construction for automata, which is also defined in Chapter II. The techniques described in Chapter II are then applied to three separate multi-UUV deployment scenarios in Chapter III. The first scenario shows two UUVs traveling at the same speed on different plans with no conflict between them. The second scenario builds on the first by introducing different operating speeds for each UUV and more complex plans. This results in more complex Moore machines. The third scenario illustrates a conflict. For this scenario, the plans result in Moore machines with many states. Only a portion of each machine is shown. A survey of related work in Chapter IV provides a review of the state of the art of multi-UUV operations today. On the surface, the work appears related, however, upon closer examination, it is not. Conclusions and future work are discussed in Chapter V.

II. FINITE-STATE MACHINES, MOORE AUTOMATA

In order to statically analyze the specification of a UUV maneuver, the maneuver has to be expressed in a way that is amenable to analysis. Conventional programming languages like C++ are far too expressive to be able to reason about their programs using a computer alone. Most questions about the behaviors of such programs are Turing undecidable. So we seek a notation whose programs can be analyzed. To this end, we adopt a type of finite-state machine called a *Moore automaton*.

A. MOORE AUTOMATA

A Moore automaton is a finite-state machine with output. More precisely, it is a tuple $M = (Q, \Sigma, \Delta, \delta, \lambda, \mathrm{s})$ where

- Q is a finite set of states,
- Σ is a finite input alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function. If *M* is in state *q* scanning input *a*, it moves to state $\delta(q, a)$,
- *s* defines the start state,
- Δ is a finite output alphabet, and
- λ maps Q to Δ giving the output associated with each state.

For example, a Moore automaton *M* that accepts precisely binary strings having 01 as a substring is given by $(Q, \Sigma, \Delta, \delta, \lambda, s)$ where

$$Q = \{q_0, q_1, q_2\}$$
$$\Sigma = \{0,1\}$$
$$\Delta = \{0,1\}$$
$$s = q_0$$

the transition function δ is defined by

$$\delta_A(q_0,0)=q_1 \quad \delta_A(q_0,1)=q_0$$
$$\delta_A(q_1,0)=q_1 \quad \delta_A(q_1,1)=q_2$$
$$\delta_A(q_2,0)=q_2 \quad \delta_A(q_2,1)=q_2$$

and the output function λ is defined by

$$\lambda(q_0)=0$$
$$\lambda(q_1)=0$$
$$\lambda(q_2)=1$$

The output for states q_0 and q_1 is 0 and for q_2 is 1. For this particular Moore machine, the output function λ is a predicate. In general, it need not be. After running on some input, *M* rests in a state for which λ is true if and only if the input contains 01 as a substring. Figure 3 illustrates an alternative way of expressing *M* as a state transition diagram. Each edge is labeled with an input symbol and every state is labeled with its name followed by the output for that state.

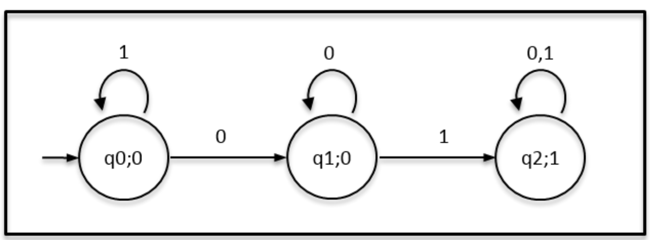

Figure 3. Example Moore Automaton Transition Diagram

A run of *M* on binary string 1110000101 is shown in Table 1. The first row shows the input string, the second row shows what state *M* is in based on the input string processed thus far, and the third row shows the output produced by each state.

Table 1. Example Moore Automaton Output

String	1	1	1	0	0	0	0	1	0	1
State	q_0	q_0	q_0	q_1	q_1	q_1	q_1	q_2	q_2	q_2
Output	0	0	0	0	0	0	0	1	1	1

B. UUV PLANS AS MOORE AUTOMATA

How is a UUV plan expressed as Moore automaton? The input alphabet becomes coordinates such as latitude and longitude. The output alphabet becomes operating instructions of the UUV such as acceleration and turning. Lastly, transitioning from a state q to a state q' on coordinate c occurs if while in state q, the UUV samples its location and finds it is described by c to within some tolerance.

For example, Figure 4 illustrates a simple UUV plan. The UUV starts at position (0, 3) and ends at position (3, 0). It has instructions to turn soft left, which it does at (2, 1), and hard right, which it does at (3, 1). These Cartesian coordinates (locations) become inputs to a Moore machine, while the instructions become outputs of the machine that the UUV must execute.

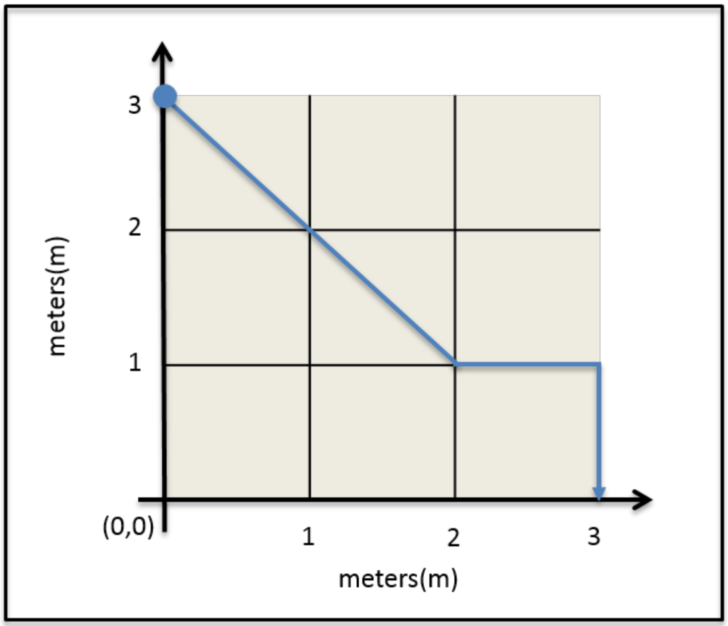

Figure 4. Example UUV Plan

A complete definition of the Moore machine for this UUV's plan is:

$$M = (Q, \Sigma, \Delta, \delta, \lambda, s)$$
$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$
$$\Sigma = \{(0,3), (1,2), (2,1), (3,1), (3,0)\}$$
$$\Delta = \{acc, idle, tsl, tr\}$$
$$s = q_0$$

the transition function δ is defined by

$$\delta(q_0, (1,2)) = q_1$$
$$\delta(q_1, (2,1)) = q_2$$
$$\delta(q_2, (3,1)) = q_3$$
$$\delta(q_3, (3,0)) = q_4$$
$$\delta(q_4, (3,0)) = q_4$$

and the output function λ is defined by

$$\lambda(q_0) = acc$$
$$\lambda(q_1) = nil$$
$$\lambda(q_2) = tsl$$
$$\lambda(q_3) = tr$$
$$\lambda(q_4) = idle$$

The output alphabet has instructions for accelerating (*acc*), do nothing (*nil*), turn soft left (*tsl*), turn hard right (*tr*) and idling in place (*idle*). The UUV in this example performs instructions based on the state it is in. In the start state q_0 the UUV performs an *acc* instruction. In state q_1 the UUV does nothing (a *nil* instruction). In state q_2, the UUV executes a *tsl* instruction and in state $q_{3,}$ a *tr* instruction. Finally, in state q_4 the UUV executes an *idle* instruction, idling the UUV in place. This last state would be in conjunction with a maneuver to the surface enabling the UUV to communicate with a host ship.

The state transition diagram for the Moore machine is shown in Figure 5.

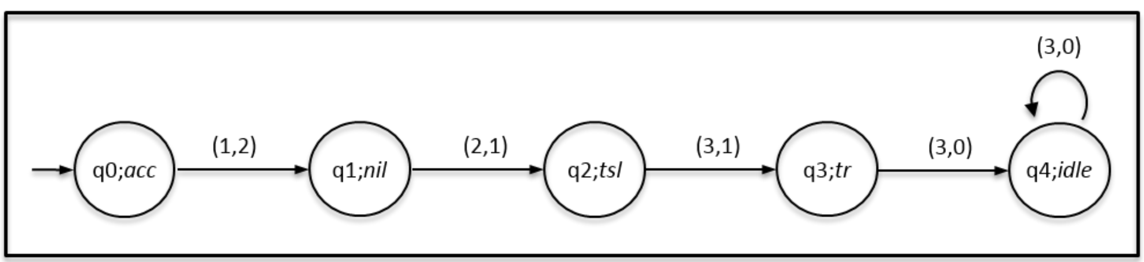

Figure 5. Example UUV Transition Diagram

C. SAMPLING RATE

Key to the coordination of multiple UUVs is that the UUVs sample their locations at the same rate. Moreover they must sample at a rate that guarantees potential collisions can always be revealed by analyzing their plans statically, that is, before they are deployed.

The sampling rate is $1/si$ where *si* is the time between samples, called the *sample interval*. Let the *Mission Operating Distance* (*MD*) denote the Euclidean distance between UUVs for safe operation (i.e., no two UUVs should be closer than *MD* from each other). The *MD* is determined strictly by the UUV's capabilities and its operators. Suppose UUV_A moves at a rate of *x*/*MD* per sample interval and UUV_B moves at a rate of *y*/*MD* per sample interval. The sample interval must be chosen so that $MD \geq x + y$. This interval is short enough to ensure UUVs sample their locations at least once to detect whether their distance apart is less than *MD* no matter how they move relative to one another at the same depth. We have $x = MD\ si\ v_1$ and $y = MD\ si\ v_2$ where v_1 and v_2 are *A* and *B*'s velocities respectively. So $x + y = MD\ si\ (v_1 + v_2)$. Now $x + y$ must not exceed *MD*, or $MD\ si\ (v_1 + v_2) \leq MD$. Thus we have $si\ (v_1 + v_2) \leq 1$, or $si \leq 1/(v_1 + v_2)$.

For given UUV velocities, the sampling rate is computed. The locations at which two UUVs are *expected* to be at *every* sample must appear in each UUV's statically analyzed plan. This does not imply that each UUV must actually sample at this rate when deployed. At run time, they need only sample frequently enough to know when to perform their instructions such as accelerating, turning or stopping. The static analysis of plans requires more frequent sampling than actually needs to occur when the UUVs are deployed. A subset of their analyzed plans is sufficient at run time. We shall see an example in Chapter V.

D. MERGING UUV PLANS

The primary advantage of using Moore automata to express UUV plans is that multiple plans can be analyzed by computer to determine, in advance of deployment, whether two UUVs deployed jointly would conflict. We define a merge operation for two plans that if successful produces a *master plan*. A master plan is merely confirmation that the two plans can operate jointly without conflict. If the merge operation fails then a conflict exists and the merge operation can pinpoint where in their plans they interfere.

The product of two plans $(\mathrm{Q}_1, \Sigma_1, \Delta_1, \delta_1, \lambda_1, s_1)$ and $(\mathrm{Q}_2, \Sigma_2, \Delta_2, \delta_2, \lambda_2, s_2)$ is the plan $(\mathrm{Q}_1 \times \mathrm{Q}_2, \Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2, \delta, \lambda, (s_1, s_2))$ where $\delta\left((p,q), A \cup B\right) = (p', q')$ if $\delta_1(p, A) = p'$, $\delta_2(q, B) = q'$, *A* is a subset of Σ_1, *B* is a subset of Σ_2 and there is no point *x* in *A* and *y* in *B* such that the Euclidean distance between *x* and *y* is less than *MD.*

The output function Λ is defined as $\Lambda(p,q) = (\lambda_1(p), \lambda_2(q))$. Examples of the merge operation are presented in Chapter III.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DEPLOYMENT SCENARIOS

This chapter includes three scenarios of joint coordination between UUVs using Moore automata. The image in Figure 6 shows the grid layout used in the scenarios. The horizontal and vertical axes are divided into increments of one nautical mile (1852 meters). For the purpose of illustrating the scenarios and the basic concept, Cartesian coordinates are used. In practice, one would use the geographic coordinate system, as defined in the ISO 19111 standard, instead.

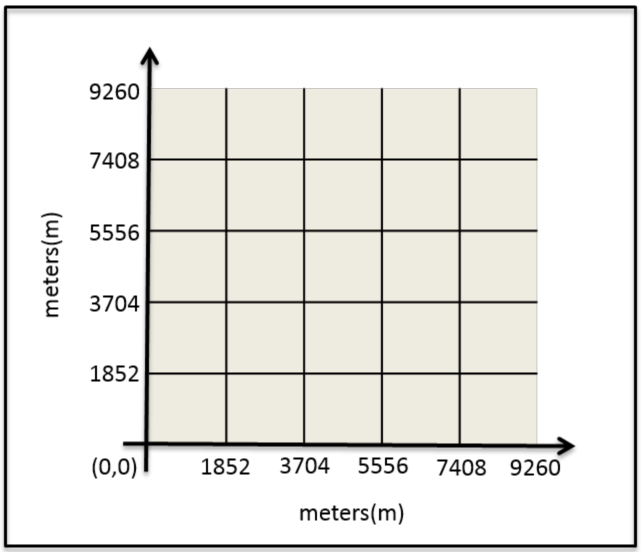

Figure 6. Grid for Defining UUV Maneuvers

A. SCENARIO ONE

In our first scenario, the goal is to define plans for the deployment of two UUVs operating jointly in Monterey Bay in support of NOAA charting and ocean floor characterization. The UUVs will surface every 30 minutes to determine their (exact) location via GPS. At other times, their location is estimated using onboard dead-reckoning. The UUVs will not surface for communication with the host ship until the mission is complete or they find themselves unexpectedly in locations not accounted for in their plans. The image in Figure 7 illustrates this scenario.

1. Sample Interval

Both UUVs will transit at 1.5 knots (.772 m/sec). Using this information we calculate their maximum sample interval:

$$\frac{1}{.772+.772} \approx 648 \text{ msec/m}.$$

Therefore the sampling rate used in their plans must be at least one sample every 648 msec in order to guarantee that any collisions can be detected statically (a priori). We choose the least sampling rate (corresponding to the maximum sample interval) to reduce the size of each UUV's plan expressed as a Moore machine. This reduces the size of the plans that are statically analyzed for potential conflicts when building a master plan for their joint operation. The illustration in Figure 7 depicts their intended paths.

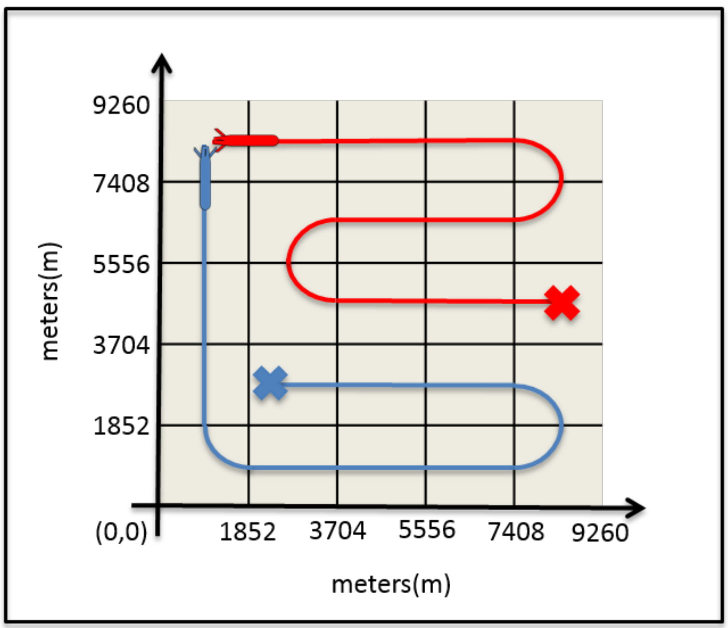

Figure 7. Scenario One

2. UUV_A Plan (Blue)

We construct the Moore machine for UUV_A using the pre-determined speed (1.5 knots) and a set of waypoints. UUV_A will travel approximately 22,224 meters and therefore the mission will take about eight hours to complete at 1.5 knots. The estimated number of samples then for UUV_A using the maximum sample interval, total distance, and distance traveled per sample becomes

$$.648\frac{\text{sec}}{\text{sample}}\times .772\frac{\text{m}}{\text{sec}}\approx .500\frac{\text{m}}{\text{sample}}\Rightarrow \frac{22,224\text{m}}{.500\frac{\text{m}}{\text{sample}}}\approx 44,627 \text{ samples}$$

At one transition per sample, this represents the number of states of the Moore machine for UUV_A. Each sample generates a unique Cartesian point that serves as an input to the Moore machine for UUV_A. In the interest of exposition, the complete 44,627-state Moore machine will not be included here but rather just portions of it to show the concept. The subset shown here will include samples every hour, coinciding with every other surface GPS positioning maneuver. Thus the portion of the Moore machine we show has only eight transitions and is defined as follows:

$$M_A=\left(Q_A,\Sigma,\Delta,\delta_A,\lambda_A,s_A\right)$$
$$Q_A=\left\{a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8\right\}$$
$$\Sigma=\begin{Bmatrix}(926,5556), & (926,2778), & (1852,926), & (4630,926), \\ (7408,926), & (8334,2778), & (5556,2778), & (2778,2778)\end{Bmatrix}$$
$$\Delta=\left\{idle,\ acc,\ nil\right\}$$
$$s_A=a_0$$

the transition function δ is defined by

$$\begin{array}{llll}
\delta_A\left(a_0,\ (926,5556)\right) & =a_1 & \delta_A\left(a_1,\ (926,2778)\right) & =a_2 \\
\delta_A\left(a_2,\ (1852,926)\right) & =a_3 & \delta_A\left(a_3,\ (4630,926)\right) & =a_4 \\
\delta_A\left(a_4,\ (7408,926)\right) & =a_5 & \delta_A\left(a_5,\ (8334,2778)\right) & =a_6 \\
\delta_A\left(a_6,\ (5556,2778)\right) & =a_7 & \delta_A\left(a_7,\ (2778,2778)\right) & =a_8 \\
\delta_A\left(a_8,\ (2778,2778)\right) & =a_8 & &
\end{array}$$

and the output function λ is defined by

$$\lambda_A(a_0) = acc$$
$$\lambda_A(a_i) = nil \text{ for } 0 < i < 8$$
$$\lambda_A(a_8) = idle$$

The output alphabet has instructions for accelerating (*acc*), do nothing (*nil*), and idling in place (*idle*). The UUV in this example performs instructions based on the state it is in. In the start state a_0 the UUV performs an *acc* instruction. In state a_1 thru a_7 the UUV does nothing. Finally, in state a_8 the UUV executes an *idle* instruction, idling the UUV in place. This last state would be in conjunction with a maneuver to the surface enabling the UUV to communicate with a host ship.

The Moore machine can also be expressed as a state transition diagram as discussed in Chapter II. The Moore machine for this example is shown in Figure 8.

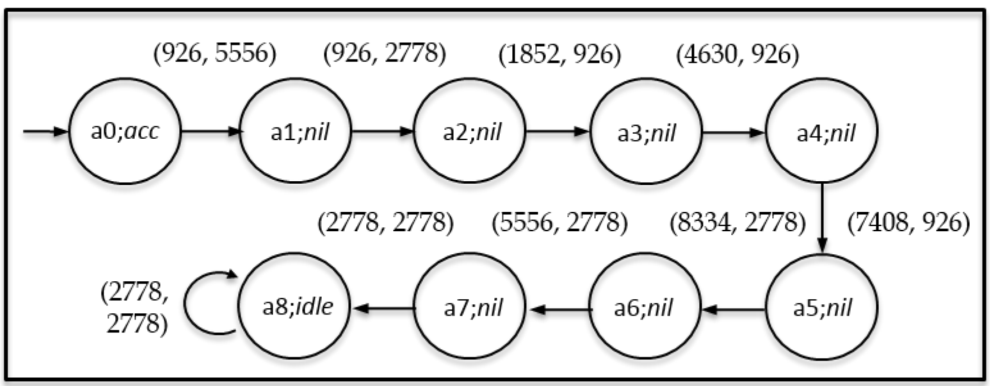

Figure 8. Subset of UUV_A's Moore Machine as a State Transition Diagram

Note that the subset of the Moore machine shown does not include all maneuvers that UUV_A expects to take when deployed. It merely illustrates a portion of the plan as a Moore machine. As such, it is insufficient as an executable machine for the UUV to run when the UUV is actually deployed. A subset of the machine for actual deployment must include all maneuvers and will be given in Chapter V.

3. UUV_B Plan (Red)

We construct the Moore machine for UUV_B using the pre-determined speed (1.5 knots) and a set of waypoints. In this scenario UUV_B will travel about the same distance as UUV_A at approximately 22,224 meters. The number of samples is also the same, at 44,627 samples. As with UUV_A, the Moore machine here represents only a portion of the complete machine. The subset of the Moore machine for UUV_B has only eight transitions and is defined as follows:

$$M_B = \left(Q_B, \Sigma, \Delta, \delta_B, \lambda_B, s_B\right)$$
$$Q_B = \left\{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\right\}$$
$$\Sigma = \begin{Bmatrix} (3704, 8334), & (6482, 8334), & (8334, 7408), & (6482, 6482), \\ (3704, 6482), & (2778, 4630), & (5556, 4630), & (8334, 4630) \end{Bmatrix}$$
$$\Delta = \left\{idle,\ acc,\ nil,\ tr,\ tl\right\}$$
$$s_B = b_0$$

the transition function δ is defined by

$$\begin{array}{ll} \delta_B\left(b_0,\ (3704, 8334)\right) = b_1 & \delta_B\left(b_1,\ (6482, 8334)\right) = b_2 \\ \delta_B\left(b_2,\ (8334, 7408)\right) = b_3 & \delta_B\left(b_3,\ (6482, 6482)\right) = b_4 \\ \delta_B\left(b_4,\ (3704, 6482)\right) = b_5 & \delta_B\left(b_5,\ (2778, 4630)\right) = b_6 \\ \delta_B\left(b_6,\ (5556, 4630)\right) = b_7 & \delta_B\left(b_7,\ (8334, 4630)\right) = b_8 \\ \delta_B\left(b_8,\ (8334, 4630)\right) = b_8 & \end{array}$$

and the output function λ is defined by

$$\lambda_B\left(b_0\right) = acc$$
$$\lambda_B\left(b_i\right) = nil \text{ for } 0 < i < 3$$
$$\lambda_B\left(b_3\right) = tr$$
$$\lambda_B\left(b_4\right) = nil$$
$$\lambda_B\left(b_i\right) = tl \text{ for } 4 < i < 7$$
$$\lambda_B\left(b_7\right) = nil$$
$$\lambda_B\left(b_8\right) = idle$$

In the start state b_0 it performs an *acc* instruction. In state b_1, b_2, b_4, and b_7 it does nothing. In state b_3 it executes a *tr* instruction. In state b_5 and b_6 it executes a *tl* instruction. Finally, in state b_8 the UUV executes an *idle* instruction, idling it in place.

The state transition diagram for this Moore machine is given in Figure 9.

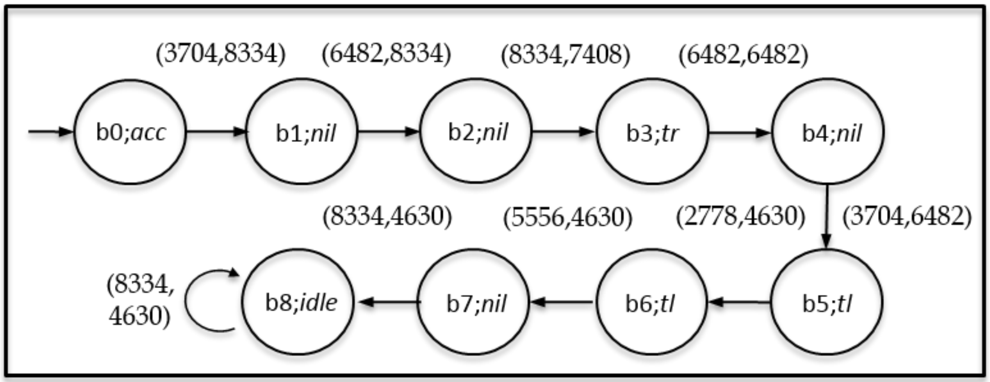

Figure 9. Subset of UUV_B's Moore Machine as a State Transition Diagram

4. Master Plan for Scenario One

With individual plans for UUV_A and UUV_B, an attempt to construct a master plan by merging their individual plans can be made. Merging guarantees absence of any conflicts during joint operation, meaning they always operate at a distance at least as large as *MD*. Therefore, merging requires a value of *MD*. Suppose for this scenario, *MD* is 400 meters. That means that the two plans cannot be merged, according to the definition of merge given in Chapter 2, if there are two states, one from each plan, from which the transitions are on Cartesian points having Euclidean distance less than 400 meters in two-dimensional space. In this scenario, the UUVs do not conflict with *MD* at 400 meters. In other words, they can be merged resulting in a master plan. The master plan as a Moore machine is defined as follows:

$$M_{AB} = \left(Q_{AB}, \Sigma, \Delta, \delta_{AB}, \lambda_{AB}, s_{AB}\right)$$

$$Q_{AB} = \left\{\left(a_0, b_0\right), \left(a_1, b_1\right), \left(a_2, b_2\right), \left(a_3, b_3\right), \left(a_4, b_4\right), \left(a_5, b_5\right), \left(a_6, b_6\right), \left(a_7, b_7\right), \left(a_8, b_8\right)\right\}$$

$$\Sigma = \left\{\begin{array}{ll} \left\{(926,5556),(3704,8334)\right\}, & \left\{(926,2778),(6482,8334)\right\}, \\ \left\{(1852,926),(8334,7408)\right\}, & \left\{(4630,926),(7408,6482)\right\}, \\ \left\{(7408,926),(3704,6482)\right\}, & \left\{(8334,2778),(2778,4630)\right\}, \\ \left\{(5556,2778),(5556,4630)\right\}, & \left\{(2778,2778),(8334,4630)\right\} \end{array}\right\}$$

$$\Delta = \left\{idle,\ acc,\ nil,\ tr,\ tl\right\}$$

$$s_{AB} = \left(a_0, b_0\right)$$

the transition function δ is defined by

$$\begin{array}{lll}
\delta_{AB}\left(\left(a_0, \mathrm{b}_0\right), \right. & \left.\left\{(926,5556),(3704,8334)\right\}\right) & = \left(a_1, \mathrm{b}_1\right) \\
\delta_{AB}\left(\left(a_1, \mathrm{b}_1\right), \right. & \left.\left\{(926,2778),(6482,8334)\right\}\right) & = \left(a_2, \mathrm{b}_2\right) \\
\delta_{AB}\left(\left(a_2, \mathrm{b}_2\right), \right. & \left.\left\{(1852,926),(8334,7408)\right\}\right) & = \left(a_3, \mathrm{b}_3\right) \\
\delta_{AB}\left(\left(a_3, \mathrm{b}_3\right), \right. & \left.\left\{(4630,926),(7408,6482)\right\}\right) & = \left(a_4, \mathrm{b}_4\right) \\
\delta_{AB}\left(\left(a_4, \mathrm{b}_4\right), \right. & \left.\left\{(7408,926),(3704,6482)\right\}\right) & = \left(a_5, \mathrm{b}_5\right) \\
\delta_{AB}\left(\left(a_5, \mathrm{b}_5\right), \right. & \left.\left\{(8334,2778),(2778,4630)\right\}\right) & = \left(a_6, \mathrm{b}_6\right) \\
\delta_{AB}\left(\left(a_6, \mathrm{b}_6\right), \right. & \left.\left\{(5556,2778),(5556,4630)\right\}\right) & = \left(a_7, \mathrm{b}_7\right) \\
\delta_{AB}\left(\left(a_7, \mathrm{b}_7\right), \right. & \left.\left\{(2778,2778),(8334,4630)\right\}\right) & = \left(a_8, \mathrm{b}_8\right) \\
\delta_{AB}\left(\left(a_8, \mathrm{b}_8\right), \right. & \left.\left\{(2778,2778),(8334,4630)\right\}\right) & = \left(a_8, \mathrm{b}_8\right)
\end{array}$$

and the output function λ is defined by

$$
\begin{aligned}
\lambda_{AB}(a_0, b_0) &= (acc, acc) \\
\lambda_{AB}(a_1, b_1) &= (nil, nil) \\
\lambda_{AB}(a_2, b_2) &= (nil, nil) \\
\lambda_{AB}(a_3, b_3) &= (nil, tr) \\
\lambda_{AB}(a_4, b_4) &= (nil, nil) \\
\lambda_{AB}(a_5, b_5) &= (nil, tl) \\
\lambda_{AB}(a_6, b_6) &= (nil, tl) \\
\lambda_{AB}(a_7, b_7) &= (nil, nil) \\
\lambda_{AB}(a_8, b_8) &= (idle, idle)
\end{aligned}
$$

In the start state (a_0, b_0) both UUVs perform an *acc* instruction. In state (a_1, b_1) and (a_2, b_2) they do nothing. In state (a_3, b_3) UUV_A does nothing and UUV_B performs a *tr* instruction. In state (a_4, b_4) they do nothing. In state (a_5, b_5) and (a_6, b_6) UUV_A does nothing and UUV_B performs a *tl* instruction. In state (a_7, b_7) they do nothing. Finally, in state (a_8, b_8) both UUVs execute an *idle* instruction, idling them in place.

The state transition diagram for the master Moore machine is shown in Figure 10.

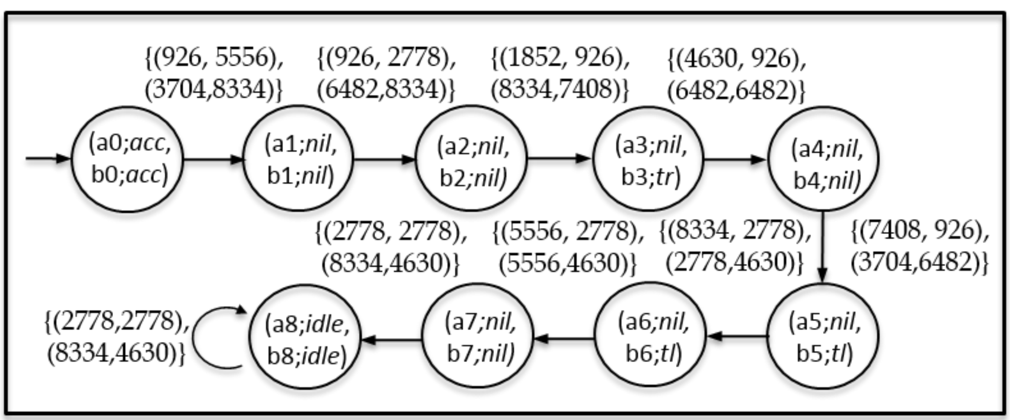

Figure 10. UUV_{AB} Master Plan as a Moore Machine

The existence of the master plan implies that unless there are unpredictable external factors, the UUVs will not come within 400 meters of each other when deployed at the same depth from their respective stated origins and headings with their stated velocities remaining constant.

B. SCENARIO TWO

In the second scenario, the goal is to define plans for the deployment of two UUVs operating jointly in search of a downed aircraft in the Gulf of Oman. The UUVs will surface every 30 minutes to determine their (exact) location via GPS. At other times their location is estimated using onboard dead-reckoning. The UUVs will not surface for communication with the host ship until the mission is complete or they find themselves unexpectedly in locations not accounted for in their plans. The image in Figure 11 illustrates this scenario.

1. Sample Interval

UUV_A will transit at 1.0 knot (.514 m/sec) while UUV_B will travel at 1.5 knots (.772m/sec). Using this information we calculate their maximum sample interval:

$$\frac{1}{.514 + .772} \approx 778 \text{ msec/m}.$$

Therefore the sampling rate used in their plans must be at least one sample every 778msec in order to guarantee that any collisions can be detected statically. Again we choose the least sampling rate (corresponding to the maximum sample interval) to reduce the size of the Moore machines analyzed in our attempt to build a master plan.

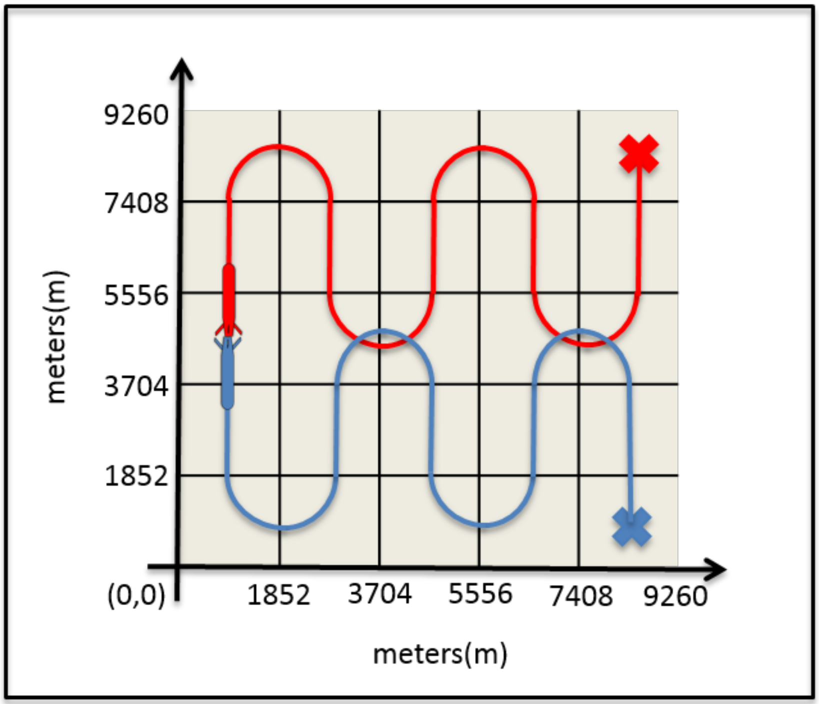

Figure 11. Scenario Two

2. UUV_A Plan (Blue)

We construct the Moore machine for UUV_A using the predetermined speed (1.0 knot) and a set of waypoints. UUV_A will travel approximately 25,928 meters and therefore the mission will take about 14 hours to complete at 1.0 knot. The estimated number of samples then for UUV_A using the sample interval, total distance traveled, and the distance traveled per sample becomes

$$.778\frac{\text{sec}}{\text{sample}}\times .514\frac{\text{m}}{\text{sec}}\approx .400\frac{\text{m}}{\text{sample}}\Rightarrow \frac{25,928\text{m}}{.400\frac{\text{m}}{\text{sample}}}\approx 64,820\text{ samples}$$

At one transition per sample, this represents the number of states of the Moore machine for UUV_A. Each sample generates a unique Cartesian point that serves as an input to the Moore machine for UUV_A. As with scenario one, the complete Moore machine will not be included here but rather a portion of it. The subset shown here will include samples every hour, coinciding with every other surface GPS positioning maneuver. Therefore, the portion of the Moore machine shown here has only 14 transitions:

$$M_A = (Q_A, \Sigma, \Delta, \delta_A, \lambda_A, s_A)$$
$$Q_A = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}\}$$
$$\Sigma = \left\{ \begin{matrix} (926, 2778), & (926, 926), & (2778, 926), & (2778, 2778), & (2778, 4630), \\ (4630, 4630), & (4630, 2778), & (4630, 2778), & (4630, 926), & (6482, 926), \\ (6482, 2778), & (6482, 4630), & (8334, 2778), & (8334, 926) & \end{matrix} \right\}$$
$$\Delta = \{idle,\ acc,\ nil,\ tr,\ tl\}$$
$$s_A = a_0$$

the transition function δ is defined by

$$\begin{array}{ll}
\delta_A(a_0,\ (926, 2778)) = a_1 & \delta_A(a_1,\ (926, 926)) = a_2 \\
\delta_A(a_2,\ (2778, 926)) = a_3 & \delta_A(a_3,\ (2778, 2778)) = a_4 \\
\delta_A(a_4,\ (2778, 4630)) = a_5 & \delta_A(a_5,\ (4630, 4630)) = a_6 \\
\delta_A(a_6,\ (4630, 2778)) = a_7 & \delta_A(a_7,\ (4630, 926)) = a_8 \\
\delta_A(a_8,\ (6482, 926)) = a_9 & \delta_A(a_9,\ (6482, 2778)) = a_{10} \\
\delta_A(a_{10},\ (6482, 4630)) = a_{11} & \delta_A(a_{11},\ (8334, 4630)) = a_{12} \\
\delta_A(a_{12},\ (8334, 2778)) = a_{13} & \delta_A(a_{13},\ (8334, 926)) = a_{14} \\
\delta_A(a_{14},\ (8334, 926)) = a_{14} &
\end{array}$$

and the output function λ

$$\begin{array}{ll}
\lambda_A(a_0) = acc & \lambda_A(a_1) = nil \\
\lambda_A(a_2) = tl & \lambda_A(a_3) = tl \\
\lambda_A(a_4) = nil & \lambda_A(a_5) = tr \\
\lambda_A(a_6) = tr & \lambda_A(a_7) = nil \\
\lambda_A(a_8) = tl & \lambda_A(a_9) = tl \\
\lambda_A(a_{10}) = nil & \lambda_A(a_{11}) = tr \\
\lambda_A(a_{12}) = tr & \lambda_A(a_{13}) = nil \\
\lambda_A(a_{14}) = idle &
\end{array}$$

In the start state a_0 it performs an *acc* instruction. In state a_1, a_4, a_7, a_{10} and a_{13} it does nothing. In state a_2, a_3, a_8 and a_9 it executes a *tl* instruction. In state a_5, a_6, a_{11} and a_{12} it executes a *tr* instruction. Finally, in state a_{14} the UUV executes an *idle* instruction, idling it in place.

The state transition diagram for the Moore machine is shown in Figure 12.

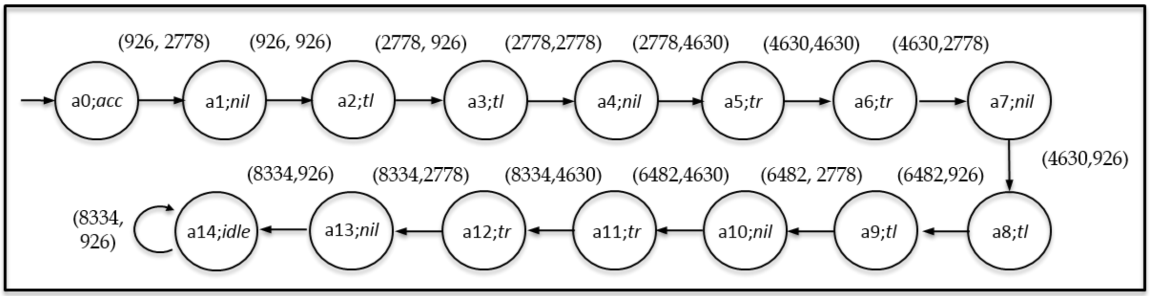

Figure 12. Subset of UUV_A's Moore Machine as a State Transition Diagram

3. UUV_B Plan (Red)

We construct the Moore machine for UUV_B using the pre-determined speed (1.5 knots) and a set of waypoints. In this scenario UUV_B will travel about the same distance as UUV_A at approximately 25,928 meters and therefore the mission will take just over 9 hours to complete at 1.5 knots. The estimated number of samples then for UUV_B using the sample interval, total distance traveled, and the distance traveled per sample becomes

$$.778\frac{\text{sec}}{\text{sample}}\times .772\frac{\text{m}}{\text{sec}}\approx .601\frac{\text{m}}{\text{sample}}\Rightarrow \frac{25{,}928\text{m}}{.601\frac{\text{m}}{\text{sample}}}\approx 43{,}142 \text{ samples}$$

At one transition per sample, this represents the number of states of the Moore machine for UUV_B. The portion of the 43,142-state machine here includes samples at every hour, coinciding with every other surface GPS positioning maneuver. Therefore, the portion of the Moore machine we show has only nine state transitions:

$$M_B = \left(Q_B, \Sigma, \Delta, \delta_B, \lambda_B, s_B\right)$$
$$Q_B = \{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9\}$$
$$\Sigma = \left\{\begin{matrix} (926,7408), & (2778,8334), & (2778,5556), & (4630,4630), & (4630,7408), \\ (6482,8334), & (6482,5556), & (8334,4630), & (8334,7408) & \end{matrix}\right\}$$
$$\Delta = \{idle,\ acc,\ tr,\ tl\}$$
$$s_B = b_0$$

the transition function δ is defined by

$$\begin{array}{ll}
\delta_B\left(b_0,\ (926,7408)\right) = b_1 & \delta_B\left(b_1,\ (2778,8334)\right) = b_2 \\
\delta_B\left(b_2,\ (2778,5556)\right) = b_3 & \delta_B\left(b_3,\ (4630,4630)\right) = b_4 \\
\delta_B\left(b_4,\ (4630,7408)\right) = b_5 & \delta_B\left(b_5,\ (6482,8334)\right) = b_6 \\
\delta_B\left(b_6,\ (6482,5556)\right) = b_7 & \delta_B\left(b_7,\ (8334,4630)\right) = b_8 \\
\delta_B\left(b_8,\ (8334,7408)\right) = b_9 & \delta_B\left(b_9,\ (8334,7408)\right) = b_9
\end{array}$$

and the output function λ is defined by

$$\begin{array}{ll}
\lambda_B(b_0) = acc & \lambda_B(b_1) = tr \\
\lambda_B(b_2) = tr & \lambda_B(b_3) = tl \\
\lambda_B(b_4) = tl & \lambda_B(b_5) = tr \\
\lambda_B(b_6) = tr & \lambda_B(b_7) = tl \\
\lambda_B(b_8) = tl & \lambda_B(b_9) = idle
\end{array}$$

In the start state b_0 it performs an *acc* instruction. In state b_1, b_2, b_5 and b_6 it executes a *tr* instruction. In state b_3, b_4, b_7 and b_8 it executes a *tl* instruction. Finally, in state b_9 the UUV executes an *idle* instruction, idling it in place.

The state transition diagram for the Moore machine is shown in Figure 13.

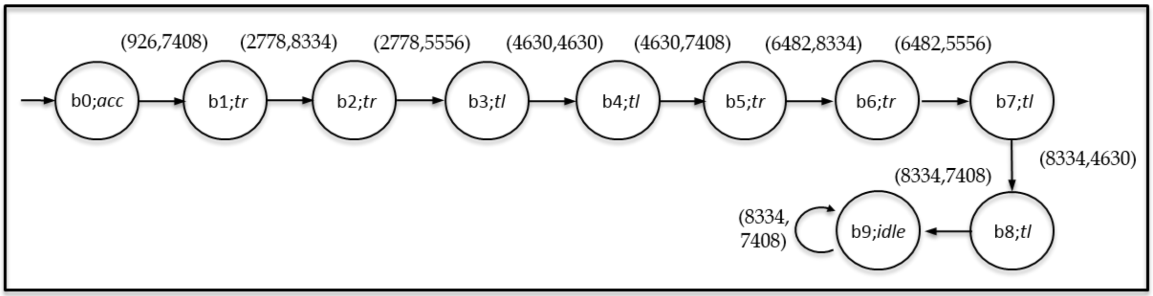

Figure 13. Subset of UUV_B's Moore Machine as a State Transition Diagram

4. Master Plan for Scenario Two

Here, much like scenario one, our goal is to guarantee there is no conflict by attempting to merge the two individual plans. For this scenario, *MD* is 200 meters. In this scenario, the UUVs do not conflict with an *MD* of 200 meters. The result is that the two plans can indeed be merged, resulting in the following master plan expressed as a Moore machine.

The master plan is:

$$M_{AB} = \left(Q_{AB}, \Sigma, \Delta, \delta_{AB}, \lambda_{AB}, s_{AB}\right)$$

$$Q_{AB} = \begin{Bmatrix} (a_0,b_0),(a_1,b_1),(a_2,b_2),(a_3,b_3),(a_4,b_4),(a_5,b_5),(a_6,b_6),(a_7,b_7), \\ (a_8,b_8),(a_9,b_9),(a_{10},b_9),(a_{11},b_9),(a_{12},b_9),(a_{13},b_9),(a_{14},b_9) \end{Bmatrix}$$

$$\Sigma = \begin{Bmatrix} \{(926,2778),(926,7408)\}, & \{(926,926),(2778,8334)\}, \\ \{(2778,926),(2778,5556)\}, & \{(2778,2778),(4630,4630)\}, \\ \{(2778,4630),(4630,7408)\}, & \{(4630,4630),(6482,8334)\}, \\ \{(4630,2778),(6482,5556)\}, & \{(4630,926),(8334,4630)\} \\ \{(6482,926),(8334,7408)\}, & \{(6482,2778),(8334,7408)\} \\ \{(6482,4630),(8334,7408)\}, & \{(8334,4630),(8334,7408)\} \\ \{(8334,2778),(8334,7408)\}, & \{(8334,926),(8334,7408)\} \end{Bmatrix}$$

$$\Delta = \{idle,\ acc,\ nil,\ tr,\ tl\}$$

$$s_B = (a_0, b_0)$$

The transition function δ is defined by

$$\delta_{AB}\left((a_0,\mathrm{b}_0),\ \{(926,2778),(926,7408)\}\right) = (a_1,\mathrm{b}_1)$$
$$\delta_{AB}\left((a_1,\mathrm{b}_1),\ \{(926,926),(2778,8334)\}\right) = (a_2,\mathrm{b}_2)$$
$$\delta_{AB}\left((a_2,\mathrm{b}_2),\ \{(2778,926),(2778,5556)\}\right) = (a_3,\mathrm{b}_3)$$
$$\delta_{AB}\left((a_3,\mathrm{b}_3),\ \{(2778,2778),(4630,4630)\}\right) = (a_4,\mathrm{b}_4)$$
$$\delta_{AB}\left((a_4,\mathrm{b}_4),\ \{(2778,4630),(4630,7408)\}\right) = (a_5,\mathrm{b}_5)$$
$$\delta_{AB}\left((a_5,\mathrm{b}_5),\ \{(4630,4630),(6482,8334)\}\right) = (a_6,\mathrm{b}_6)$$
$$\delta_{AB}\left((a_6,\mathrm{b}_6),\ \{(4630,2778),(6482,5556)\}\right) = (a_7,\mathrm{b}_7)$$
$$\delta_{AB}\left((a_7,\mathrm{b}_7),\ \{(4630,926),(8334,4630)\}\right) = (a_8,\mathrm{b}_8)$$
$$\delta_{AB}\left((a_8,\mathrm{b}_8),\ \{(6482,926),(8334,7408)\}\right) = (a_9,\mathrm{b}_9)$$
$$\delta_{AB}\left((a_9,\mathrm{b}_9),\ \{(6482,2778),(8334,7408)\}\right) = (a_{10},\mathrm{b}_9)$$
$$\delta_{AB}\left((a_{10},\mathrm{b}_9),\ \{(6482,4630),(8334,7408)\}\right) = (a_{11},\mathrm{b}_9)$$
$$\delta_{AB}\left((a_{11},\mathrm{b}_9),\ \{(8334,4630),(8334,7408)\}\right) = (a_{12},\mathrm{b}_9)$$
$$\delta_{AB}\left((a_{12},\mathrm{b}_9),\ \{(8334,2778),(8334,7408)\}\right) = (a_{13},\mathrm{b}_9)$$
$$\delta_{AB}\left((a_{13},\mathrm{b}_9),\ \{(8334,926),(8334,7408)\}\right) = (a_{14},\mathrm{b}_9)$$
$$\delta_{AB}\left((a_{14},\mathrm{b}_9),\ \{(8334,926),(8334,7408)\}\right) = (a_{14},\mathrm{b}_9)$$

and the output function λ is defined by

$$\lambda_{AB}(a_0,b_0) = (acc,acc) \qquad \lambda_{AB}(a_1,b_1) = (nil,tr)$$
$$\lambda_{AB}(a_2,b_2) = (tl,tr) \qquad \lambda_{AB}(a_3,b_3) = (tl,tl)$$
$$\lambda_{AB}(a_4,b_4) = (nil,tl) \qquad \lambda_{AB}(a_5,b_5) = (tr,tr)$$
$$\lambda_{AB}(a_6,b_6) = (tr,tr) \qquad \lambda_{AB}(a_7,b_7) = (nil,tl)$$
$$\lambda_{AB}(a_8,b_8) = (tl,tl) \qquad \lambda_{AB}(a_9,b_9) = (tl,idle)$$
$$\lambda_{AB}(a_{10},b_9) = (nil,idle) \qquad \lambda_{AB}(a_{11},b_9) = (tr,idle)$$
$$\lambda_{AB}(a_{12},b_9) = (tr,idle) \qquad \lambda_{AB}(a_{13},b_9) = (nil,idle)$$
$$\lambda_{AB}(a_{14},b_9) = (idle,idle)$$

The state transition diagram for the master Moore machine is shown in Figure 14.

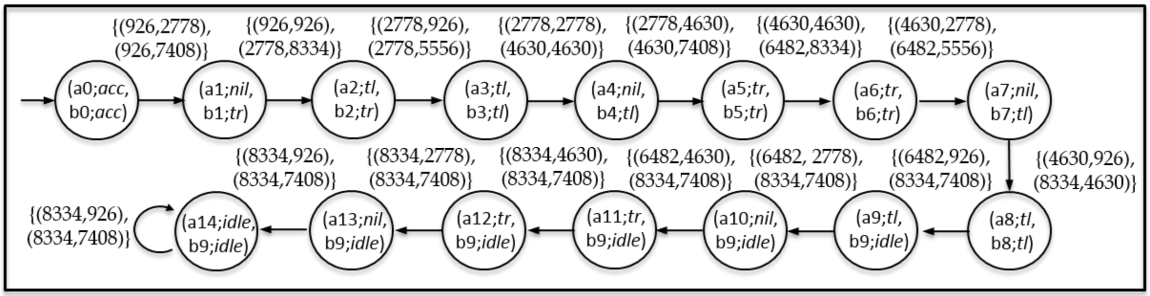

Figure 14. UUV_{AB} Moore Machine Master Plan

C. SCENARIO THREE

This scenario is designed specifically to show conflict. The operating area is reduced to 1852 m^2 to allow for computation of the sample set for each UUV of which a portion is included in the Appendix. The image in Figure 15 illustrates this scenario.

1. Sampling Interval

For this scenario UUV_A will transit at 2.0 knots (1.03 m/sec) while UUV_B will transit at 1.0 knot (.514 m/sec). Using this information we calculate their maximum sample interval:

$$\frac{1}{1.03+.514} \approx 648\text{msec/m}\,.$$

Therefore, the sampling rate used in their plans must be at least one sample every 648msec in order to guarantee that any collisions can be detected statically. Again we choose the least sampling rate to reduce the size of the Moore machines for static analysis.

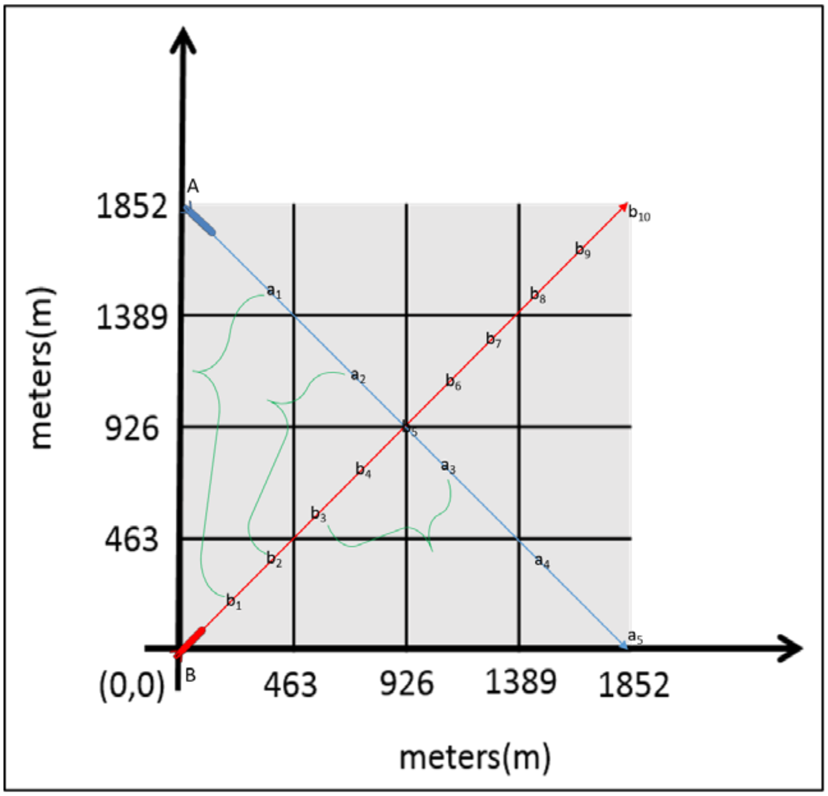

Figure 15. Scenario Three

2. UUV_A Plan (Blue)

We construct the Moore machine for UUV_A using the pre-determined speed (2.0 knots) and a set of waypoints. UUV_A will travel approximately 2,619 meters therefore the mission will take about 40 minutes to complete at 2.0 knots. The estimated number of samples then for UUV_A using the sample interval, total distance, and distance traveled per sample becomes

$$.648\frac{\text{sec}}{\text{sample}}\times 1.03\frac{\text{m}}{\text{sec}} \approx .667\frac{\text{m}}{\text{sample}} \Rightarrow \frac{2,619\text{m}}{.667\frac{\text{m}}{\text{sample}}} \approx 3,927 \text{ samples}$$

At one transition per sample, this represents the number of states of the Moore machine for UUV_A. A subset of the 3,927-state machine is shown using eight minute samples:

$$M_A = (Q_A, \Sigma, \Delta, \delta_A, \lambda_A, s_A)$$
$$Q_A = \{a_0, a_1, a_2, a_3, a_4, a_5\}$$
$$\Sigma = \left\{ \begin{matrix} (346.45, 1505.55), & (692.44, 1159.57), & (1038.89, 813.11), \\ (1384.87, 467.13), & (1731.32, 120.68) & \end{matrix} \right\}$$
$$\Delta = \{idle,\ acc,\ nil\}$$
$$s_A = a_0$$

the transition function δ is defined by

$$\delta_A(a_0,\ (346.45, 1505.55)) = a_1$$
$$\delta_A(a_1,\ (692.44, 1159.57)) = a_2$$
$$\delta_A(a_2,\ (1038.89, 813.11)) = a_3$$
$$\delta_A(a_3,\ (1384.87, 467.13)) = a_4$$
$$\delta_A(a_4,\ (1731.32, 120.68)) = a_5$$
$$\delta_A(a_5,\ (1731.32, 120.68)) = a_5$$

and the output function λ is defined by

$$\lambda_A(a_0) = acc$$
$$\lambda_A(a_i) = nil \text{ for } 0 < i < 5$$
$$\lambda_A(a_5) = idle$$

For this plan, the Cartesian coordinates are taken from those listed in the Appendix. The state transition diagram for the Moore machine is shown in Figure 16.

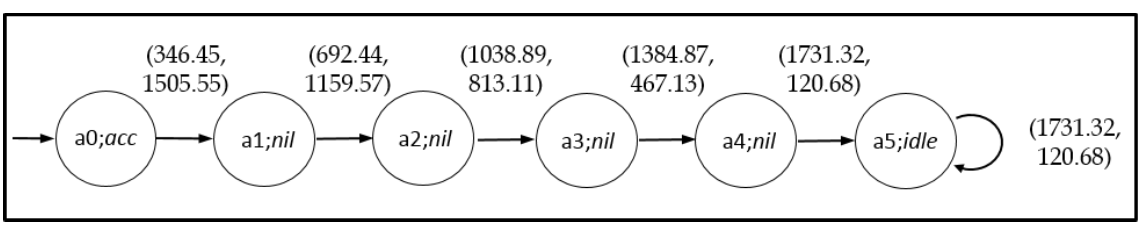

Figure 16. Subset of UUV_A's Moore Machine as a State Transition Diagram

3. UUV_B Plan (Red)

We construct the Moore machine for UUV_B using the pre-determined speed (1.0 knot) and a set of waypoints. UUV_B will travel approximately 2,619 meters and therefore the mission will take about 80 minutes to complete at 1.0 knot. The estimated number of samples then for UUV_B using the sample interval, total distance, and distance traveled per sample becomes

$$.648\frac{\text{sec}}{\text{sample}}\times .514\frac{\text{m}}{\text{sec}}\approx .333\frac{\text{m}}{\text{sample}}\Rightarrow \frac{2,619\text{m}}{.333\frac{\text{m}}{\text{sample}}}\approx 7,865\text{ samples}$$

At one transition per sample, this represents the number of states of the Moore machine for UUV_A. A subset of the complete 7,865-state machine is shown below using samples every 8 minutes:

$$M_B=\left(Q_B,\Sigma,\Delta,\delta_B,\lambda_B,s_B\right)$$
$$Q_B=\left\{b_0,b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9,b_{10}\right\}$$
$$\Sigma=\left\{\begin{matrix}(174.53,174.53),(348.83,348.83),\\(523.36,523.36),(697.66,697.66),\\(872.19,872.19),(1046.49,1046.49),\\(1221.02,1221.02),(1395.31,1395.31),\\(1569.85,1569.85),(1744.14,1744.14)\end{matrix}\right\}$$
$$\Delta=\left\{idle,\ acc,\ nil\right\}$$
$$s_B=b_0$$

the transition function δ is defined by

$$
\begin{array}{llll}
\delta_B\left(b_0,\ (174.53,174.53)\right) & = b_1 & \delta_B\left(b_1,\ (348.83,348.83)\right) & = b_2 \\
\delta_B\left(b_2,\ (523.36,523.36)\right) & = b_3 & \delta_B\left(b_3,\ (697.66,697.66)\right) & = b_4 \\
\delta_B\left(b_4,\ (872.19,872.19)\right) & = b_5 & \delta_B\left(b_5,\ (1046.49,1046.49)\right) & = b_6 \\
\delta_B\left(b_6,\ (1221.02,1221.02)\right) & = b_7 & \delta_B\left(b_7,\ (1395.31,1395.31)\right) & = b_8 \\
\delta_B\left(b_8,\ (1569.85,1569.85)\right) & = b_9 & \delta_B\left(b_9,\ (1744.14,1744.14)\right) & = b_{10} \\
\delta_B\left(b_{10},\ (1744.14,1744.14)\right) & = b_{10} & &
\end{array}
$$

and the output function λ defined by

$$
\begin{aligned}
\lambda_B(b_0) &= acc \\
\lambda_B(b_i) &= nil \text{ for } 0 < i < 10 \\
\lambda_B(b_{10}) &= idle
\end{aligned}
$$

The state transition diagram for the Moore machine is shown in Figure 17.

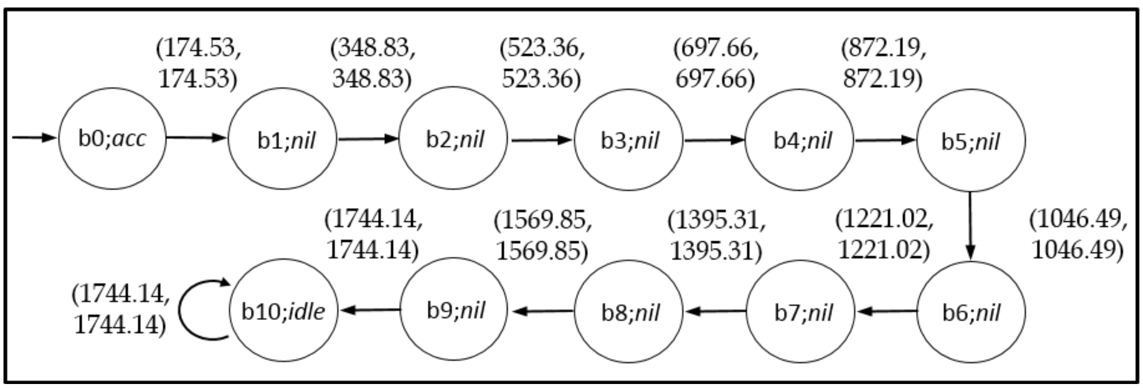

Figure 17. Subset of UUV_B's Moore Machine as a State Transition Diagram

4. Master Plan for Scenario Three

Here we attempt to construct a master plan with *MD* equal to 600 meters. Unlike scenarios one and two, when we attempt to merge the individual plans for UUV_A and UUV_B to create the master plan, a conflict is detected. Hence no master plan exists. However, a partial master plan can be constructed to reveal where the conflict arises. A partial state transition diagram for the master Moore machine is shown in Figure 18.

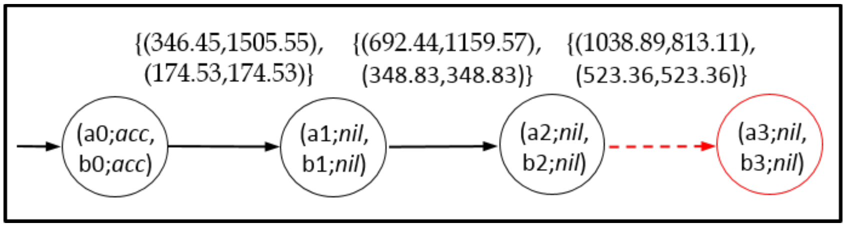

Figure 18. Partial Moore Machine for Partial Master Plan

Only three states of the master plan are possible, as illustrated in Figure 18. There is a conflict when attempting to transition out of state (a_2, b_2). The Appendix shows the samples for state (a_2, b_2) and (a_3, b_3) to be 1477 and 2216, respectively. In state (a_2, b_2) the distance between the two UUVs is approximately 880 meters, which does not violate *MD*; however, in state (a_3, b_3) the distance between them drops to approximately 591 meters, which violates *MD*. Violation of *MD* actually occurred at sample 2164, roughly 23 minutes into their planned transits. The graph in Figure 19 shows the total distance traveled by each UUV and the distance between them at each sample, had there been no conflict. After UUV_A completes its mission, it sits idle while UUV_B completes its mission. Although not readily apparent, the green line showing the distance between them dips below *MD*.

The ability to identify the point where conflict occurs is another benefit of our approach. It suggests ways to resolve the conflict, for example, by adjusting *MD* if possible, adjusting speed, or selecting alternative waypoints. Once the adjustments are made, another attempt can be made to construct a master plan.

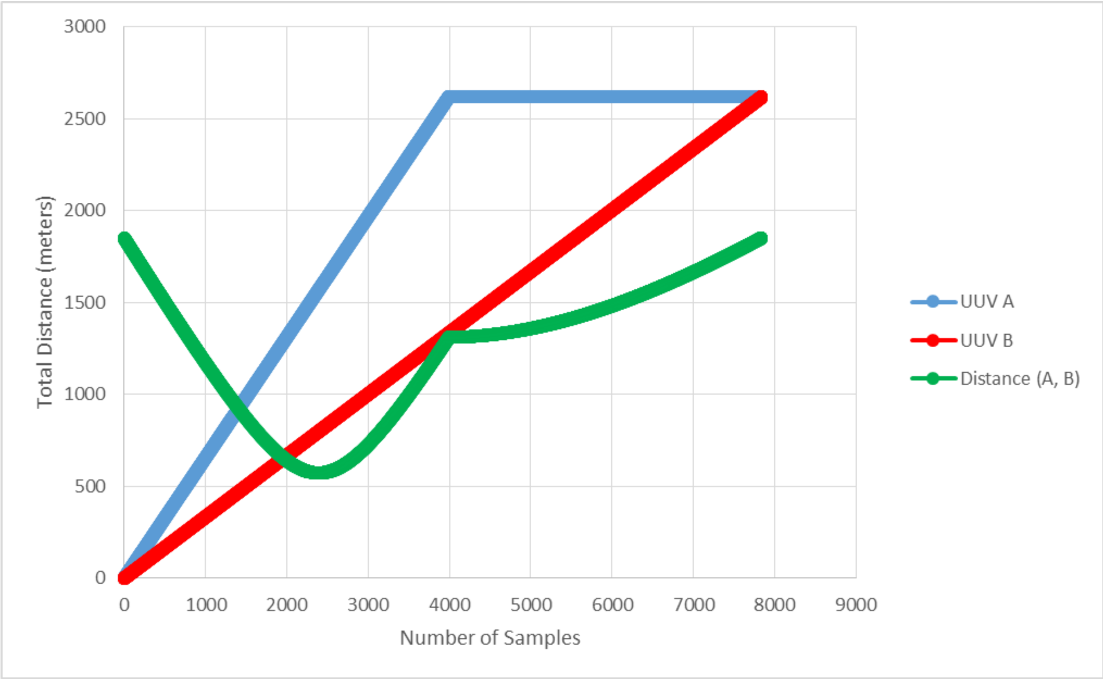

Figure 19. Scenario Three Distance Graph

IV. A SURVEY OF RELATED WORK

As mentioned in Chapter I, our approach is aimed at rapid deployment of multiple UUVs without relying on any communication between them. Much of the work in joint UUV deployment relies on communication. As such, it is not well suited for the type of deployment we aim to address: a coalition marshaling UUVs on short notice to operate jointly without communication. Nevertheless, it is instructive to understand some of this work. Some kind of hybrid approach may be needed to address individual UUV recovery plans, which are a source of difficulty; see Chapter V.

A. COMMUNICATION-BASED UUV OPERATIONS

Ouimet [2] expanded on previous work with undersea acoustic networking technology developed by the Navy to navigate Slocum glider UUVs. In his experiments it is evident that environmental factors and underwater sound propagation issues all contributed to the high error rates observed. It was intended as a fundamental step towards UUV swarm collaboration; however, there appears to be no follow-up to his work.

Realizing the limitation on providing the warfare commander with near real-time data from AUVs on station, Marr [6] proposed and simulated rendezvous capabilities between multiple AUVs. A "searcher" AUV acoustically offloaded data to a "server" AUV that then surfaced and transmitted data to the warfare commander via radio frequency (RF) or satellite link. This method allowed the "searcher" AUV to remain on station to provide continuous support. Much like the previous example, Marr realized underwater sound propagation severely limited the acceptable distances for transferring data between the two AUVs.

Nicholson's [5] efforts demonstrated partial implementation of Marr's [6] work. Nicholson proposed rendezvous capabilities between multiple AUVs using the acoustic radio interactive exploratory server (ARIES) AUV. Much like [6], to maximize time-on-station in data-gathering AUVs, deployment of a "server" AUV was tasked with downloading data from a data-gathering AUV, minimizing downtime and maximizing data collection. Nicholson demonstrated successful rendezvous and communications between an ARIES AUV (server) and a pre-programmed virtual AUV (data-gatherer).

While their efforts demonstrate some form of coordination, it also highlights the inherent weaknesses in sound propagation. This thesis demonstrated a new method for UUV coordination without the use of any underwater communications. The remainder of this section discusses three research projects that were effective in the coordination of multiple UUVs without any, or very little, communications between them.

B. A TWO-LEVEL, PROTOCOL-BASED APPROACH TO CONTROLLING AUTONOMOUS OCEANOGRAPHIC SAMPLING NETWORKS

Turner [4] presented an approach to adapt the existing autonomous oceanographic sampling networks (AOSN) construct to handle more complex mission control for UUVs. His cooperative distributed AOSN, or *CoDA*, was a project that focused on intelligent control mechanisms for advanced AOSNs. AOSNs are discussed in detail in [8]. AOSNs were originally developed to advance the state-of-the-art in understanding ocean characteristics. Turner identified a need for better network management given the myriad of environmental and mechanical factors involved in maintaining an underwater network for an extended period of time. Turner's work was unique in that it introduced a self-sufficient hierarchy into the network by creating the meta-level organization (MLO) and task-level organization (TLO). The MLO determined the capability of the AOSN as a whole; this included all nodes and UUVs involved, taking into account their capabilities and limitations. The MLO addressed extended operations by dynamically reorganizing the network based on the environment; whether due to vehicles coming or leaving the network, vehicles failing, or other environmental factors. The TLO then assigned tasks based the MLO input. This project was platform agnostic and tasked the inquiring AUV based on its capabilities, identified during initial deployment into the network. If a tasked UUV failed during operation, the network would dynamically reorganize and reassign missions to other capable AUVs.

While Turner’s work demonstrated coordination among AUVs and an underwater network, it did so while relying on underwater communications. Additionally, the hierarchical setup of his network required that the UUVs be aware of one another.

C. COORDINATED CONTROL OF MULTIPLE AUTONOMOUS UNDERWATER VEHICLE SYSTEMS

D. Jiang et al. [9] builds on Botelho’s [10] robot coordination concept and applies it to UUV coordination. D. Jiang et al. used the mission-oriented operating suite interval programming (MOOS-IvP) architecture (open source) and a market-based approach to fully realize distributed control of underwater vehicles with tightly coupled actions. P. Newman [11] defined MOOS as:

> A set of libraries and applications designed to facilitate research in the mobile robotic domain. The spectrum of functionality provided ranges over low-level, multi-platform communications, dynamic control, high precision navigation and path planning, concurrent mission task arbitration and execution, mission logging and playback.

According to [9], MOOS functioned as a suite of software modules that coordinate software processes running on an autonomous platform. The IvP was a technique for solving multi-objective optimization problems.

In [9], an auctioneer AUV declared a set of tasks, and each AUV calculated its cost to execute those tasks. Each AUV carried its own standardized database, which it then communicated using a unified communication interface—called MOOSBridge—to a central database that housed similar information from other AUV subscribers. This central database fed the necessary information back individual subscribers, making them aware of other taskings, allowing the group to continue in a coordinated fashion. Whichever AUV submitted the lowest bid would win that auction. This construct required underwater communications between the AUVs and a central database. However, the AUVs were programmed to limit communications due to sound propagation concerns. Simulations for this project showed the behavior of AUVs operating as intended, as seen in Figure 20. The objective of this simulation was to visit each station while minimizing distance traveled between all three AUVs.

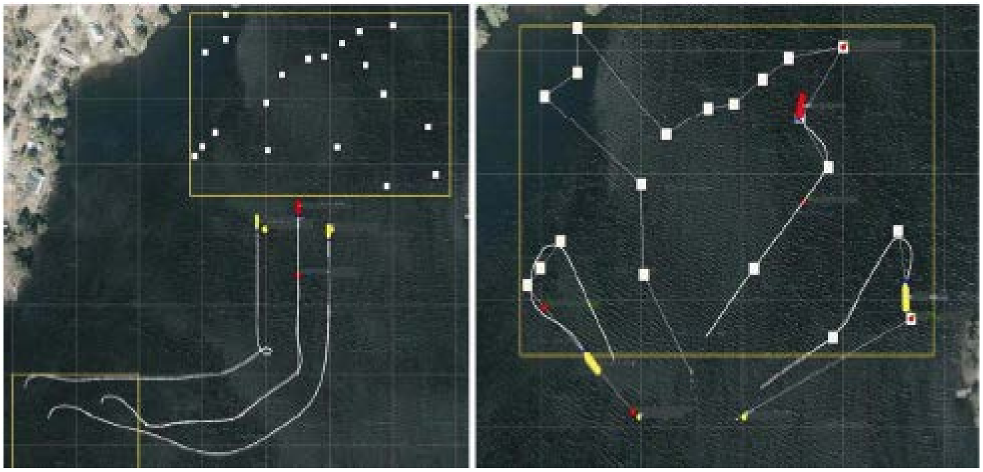

Figure 20. MOOS-IvP Simulation Test Run using the pMarineViewer Graphical User Interface, from [9]

Some of the weaknesses in this project were as follows. Although D. Jiang et al. [9] used their MOOSBridge module to simulate the unified communications between the vehicles, they did so over a local area network (LAN). This eliminated demonstrating the biggest constraint—underwater communications. Additionally, their simulations did not indicate whether their network retained the ability to adapt to a loss of resources, for example, an AUV failure. While this project demonstrated both coordinated and independent movement of AUVs, it did so using underwater communications.

D. MULTI-AUV CONTROL AND ADAPTIVE SAMPLING IN MONTEREY BAY

Fiorelli et al. [12] described a process for multi-AUV control using virtual bodies and artificial potentials (VBAP). This project is of particular interest because it managed multi-AUV control without the use of underwater communications. Artificial potential fields (APFs) were used to drive the autonomous vehicles toward a desired goal, or end state. The use of APFs also enabled autonomous formation control between the cooperating AUVs. More information on APFs can be found in [13].

The virtual body introduced the mission to the group of AUVs, synchronizing formation control efforts with the desired task. Waypoint lists were generated using VBAP output and transmitted to the gliders via an Iridium connection. During this experiment, the AUVs surfaced every two hours to receive mission updates using previously uploaded data from the lead AUV. Through intensive human intervention (every two hours), the formations were able to operate autonomously towards a goal without the use of underwater communications. This project advanced existing theory by demonstrating the use of VBAP in multi-AUV control. Additional contributions in this work included the ability of the formation to adjust its mission based on real-time sampling results. While this was nothing new for an individual AUV, it was new for a group of AUVs to move in concert based on sample data. Some of the weaknesses in this project are as follows:

- Gliders, in general, are slow and average a constant speed due to the nature of their design. This made it difficult for AUVs to maintain formation when external factors, such as current, weather, etc., were affecting their ability to navigate. It also inhibited implementation of formation control using artificial potentials.

- Ideally, the goal was to use the most recent offloaded data to generate the next two-hour mission interval; however, to minimize the time AUVs spent on the surface, data from the previous cycle was used instead.

- The latest mission was uploaded to the lead AUV, and then it resumed operations. This limited the ability of the operator to upload newer, more recent or accurate data to the subsequent surfacing AUVs. Had the operator done so, the AUVs would no longer have been synchronized with the lead glider.

While demonstrating multi-AUV coordination without underwater communication, Fiorelli et al. [12] did so at the expense of near continuous human-in-the-loop support, limiting the deployment feasibility in a real world scenario. Additionally, there was no demonstration of independent AUV operation using the VBAP construct. This thesis, by contrast focuses on the ability to rapidly deploy multiple AUVs, independently with little to no preparation, very little human-in-the-loop intervention and no underwater communications.

One of the fundamental capabilities absent from these examples is rapid deployment. The term rapid is synonymous with nearly every capability listed in the Navy's master plan [1]. Additionally, there needs to be another way to ensure success in coordination without relying on underwater communications. Lastly, the requisite manpower required for the coordination demonstrated in these examples is impractical.

V. CONCLUSIONS AND FUTURE WORK

This thesis presents a new approach to autonomous underwater vehicle navigation based on the static analysis of their plans. Autonomous vehicles are normally controlled by software expressed in a control language that defies mechanical processing. Consequently, the software cannot be easily, if at all, analyzed by machine to determine whether there may be conflicts among UUV plans. This thesis proposes a new technique for expressing plans, called Moore automata. These automata are more amenable to static analysis for detecting conflicts.

There are aspects of the new approach that require attention, notably, considerations for its practical use and its inherent limitations. This chapter addresses both.

A. PRACTICAL CONSIDERATIONS

1. Static versus Runtime Plans

The velocities of UUVs determine a maximum sample interval. It gives the smallest sampling rate for the UUVs; however, the rate can cause a large number of states in the Moore automata describing their plans. We call these plans their *static plans* to distinguish them from the plans they actually run when deployed, which we call their *runtime plans*. If the static plans have a master plan then there is no conflict for the chosen *MD*. From the static plans one can extract a runtime plan based on an even smaller sampling rate, giving rise to far fewer states in the Moore machine. The only requirement is that all maneuvers in the static plan be preserved in the runtime plan.

For example, we can extract a runtime plan from the Moore machine static plan for UUV_A of the first scenario in Chapter III Its definition is given as follows:

$$M_A = (Q_A, \Sigma, \Delta, \delta_A, \lambda_A, s_A)$$
$$Q_A = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$$
$$\Sigma = \begin{Bmatrix} (926,1852), & (1852,926), & (7408,926), \\ (8334,1852), & (7408,2778), & (2778,2778) \end{Bmatrix}$$
$$\Delta = \{idle, acc, nil, tsl\}$$
$$s_A = a_0$$

The state transition diagram for UUV_A's runtime plan is shown in Figure 21.

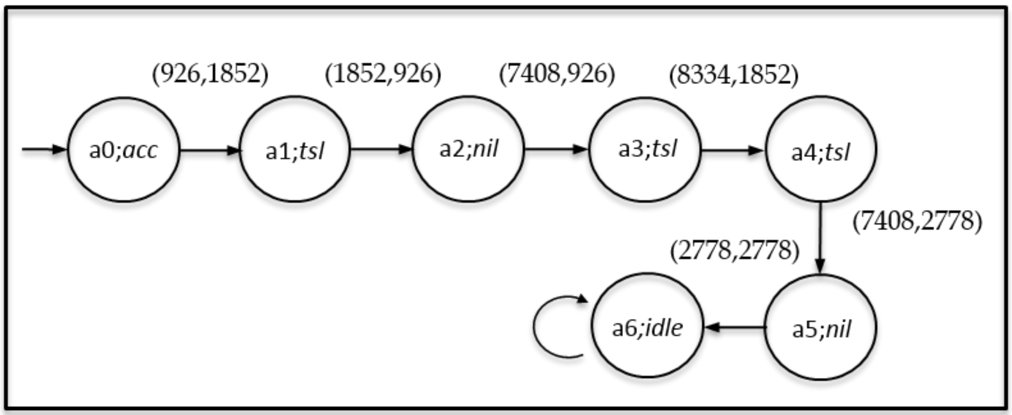

Figure 21. State Transition Diagram for Scenario One Runtime Plan

The resulting runtime plan consists of seven states, and preserves the output of UUV_A which is necessary for maneuvers. This is far fewer than the 44,627 states required for static analysis.

2. Tolerance

Interference is detected during static analysis by computing the Euclidean distance between two coordinates in two-dimensional space. If weather conditions or other factors suggest that UUVs might deviate from their plans when deployed, then one can introduce an error tolerance into the analysis. However, this makes the detection of a collision more likely with the same input parameters, depending on the tolerance. Favorable conditions, for example, sea state, weather, and shallow or deep operations, would call for a smaller tolerance. The greater the error tolerance, the more likely there is a collision detected by the static analysis.

For instance, suppose conditions support a 5% error tolerance and let *MD* be 400 meters. Tolerance then becomes $400(.05) = 20 \text{ meters}$. We replace *MD* with $MD_{\text{tolerance}}$ where $MD_{\text{tolerance}}$ is defined as $MD + 2r = 440 \text{ meters}$. $MD_{\text{tolerance}}$ reflects the worst-case situation, in which two UUVs are operating at the edge of their tolerance closest to each other. If conditions were less favorable, we might use a tolerance of 15%; this results in an acceptable radius of 60 meters for each UUV and an $MD_{\text{tolerance}}$ of 520 meters.

3. Static Analysis for Three Dimensions

UUVs operate in a three dimensional environment, and although this thesis limited the analysis to two dimensions, little additional work is needed to apply these results to a three dimensional environment. UUVs, particularly gliders, operate at fixed depth windows called a yo-yo. You can set a max depth and max ceiling to define a yo-yo. De-conflicting yo-yo ceiling and depth figures during static analysis would resolve any concerns over conflict in that third dimension. Furthermore, Euclidean distance in three dimensions is also well defined.

4. More than Two UUVs

If the plan is to deploy more than two UUVs then their static plans must agree on a sampling rate. How is this rate determined? The sound approach is to take the worst-case sampling rate corresponding to the two fastest UUVs assuming they are heading directly for each other, even though they may have no intention of doing so. Let v_i and v_j be the two fastest velocities in a group of UUVs (v_1, v_2,…,v_n), then

$$\frac{1}{v_i + v_j}$$

becomes the desired sample interval.

B. INHERENT LIMITATIONS

As this approach is based on static analysis, it is attempting to say something about runtime behavior by examining only static information. The static analysis is precise to the extent that the information it uses remains static. If after a static analysis has determined there is a master plan for two UUVs, but then the UUVs don't follow their extracted runtime plans when deployed for whatever reason—perhaps beyond their control—then the analysis is useless. This is true of any static analysis. However, it is conjectured that there are many useful situations in which information about UUV navigation does remain static so static analysis wins. But there are other situations in which it does not. We look at two of them here.

1. Changing Velocity during Runtime

If a UUV determines during runtime that it needs to speed up based on information not available during the static analysis, such as speed of the current, then there is no way to account for this a priori. Accounting for changes in speed in the static analysis is only possible if these changes are known at the time of the analysis. Otherwise, any master plan constructed is meaningless even if operating conditions are predictable.

2. Recovery from Unpredictable Conditions

Inevitably, at some point during operation a UUV will find itself in an unexpected location, due perhaps to changes in current, weather or some other external factor. This location may not be accounted for in the UUV's runtime plan. A UUV can have as part of its plan some recovery maneuvers. But a problem arises if one tries to construct a master plan for multiple UUVs and each UUV has recovery maneuvers.

For instance, say an attempt to merge the plans of two UUVs reveals a conflict because of an attempt to transition on the same Cartesian point. To resolve the conflict one of the UUVs must not take that transition, and instead must transition to another conditional state; however, there is no way to account for the conditional state in the static plan because there is no way to know a priori what factors caused the UUV to arrive at that state. This example is illustrated in Figure 22. The transition diagram (a) represents a UUV that will transition on the coordinates (x_1, y_1). The transition diagram (b) represents a UUV that also intends to transition on the coordinates (x_1, y_1); however, it also has a transition on (x_2, y_2) just in case it finds itself in this location for reasons beyond its control. It then tries to recover with a *tr* instruction. In an attempt to merge the plans, the only option the UUV in (b) has is to transition on (x_2, y_2) but it does not know how to reach that location because it's a location reached beyond its control. No such location can be part of any master plan.

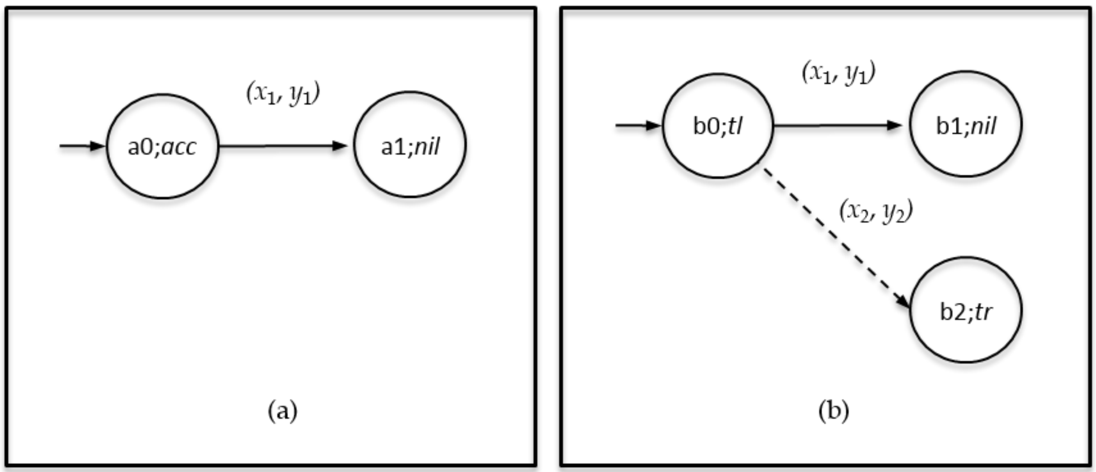

Figure 22. Reproducing an Unpredictable Condition

C. FUTURE WORK

1. Planned Changes in Velocity

In order to treat changing velocities, consider the case in which two UUVs have multiple speed adjustments built into their plans. Each adjusts its speed independently. A new sampling rate must be calculated for every interval where an interval is defined to be the time from when a UUV changes its speed until it or the other UUV changes its speed, if ever again. An attempt is made to construct a master plan for each interval. If successful, then there is a master plan for the entire mission.

2. Recovery from Unpredictable Conditions

An important area of future work is coping with recovery steps that a UUV takes in response to unforeseen operating conditions. A recovery maneuver cannot be part of any master plan because it is taken due to circumstances beyond the UUV's control. Further, a UUV might try to recover while in a recovery! Currently, no recovery paths can be part of any plan. This is where a hybrid approach may be useful, one that combines static analysis with some runtime communications just in the event a UUV finds it has deviated from its plan.

APPENDIX. SCENARIO THREE CALCULATIONS

				Vehicle A		Vehicle B		Distance (A, B)
Sample #	Time(sec)	Dist A trvl (m)	Dist B trvl (m)	x1	y1	x2	y2	
1457	947.05	965.991	486.638	683.0587867	1168.941213	344.1050298	344.1050298	891.7647554
1458	947.7	966.654	486.972	683.5275985	1168.472402	344.3412034	344.3412034	891.2012355
1459	948.35	967.317	487.306	683.9964103	1168.00359	344.5773771	344.5773771	890.6379779
1460	949	967.98	487.64	684.4652221	1167.534778	344.8135508	344.8135508	890.074983
1461	949.65	968.643	487.974	684.9340338	1167.065966	345.0497244	345.0497244	889.5122514
1462	950.3	969.306	488.308	685.4028456	1166.597154	345.2858981	345.2858981	888.9497835
1463	950.95	969.969	488.642	685.8716574	1166.128343	345.5220718	345.5220718	888.3875799
1464	951.6	970.632	488.976	686.3404692	1165.659531	345.7582454	345.7582454	887.825641
1465	952.25	971.295	489.31	686.809281	1165.190719	345.9944191	345.9944191	887.2639674
1466	952.9	971.958	489.644	687.2780928	1164.721907	346.2305928	346.2305928	886.7025595
1467	953.55	972.621	489.978	687.7469046	1164.253095	346.4667664	346.4667664	886.1414179
1468	954.2	973.284	490.312	688.2157164	1163.784284	346.7029401	346.7029401	885.5805431
1469	954.85	973.947	490.646	688.6845282	1163.315472	346.9391138	346.9391138	885.0199355
1470	955.5	974.61	490.98	689.15334	1162.84666	347.1752874	347.1752874	884.4595957
1471	956.15	975.273	491.314	689.6221518	1162.377848	347.4114611	347.4114611	883.8995242
1472	956.8	975.936	491.648	690.0909636	1161.909036	347.6476348	347.6476348	883.3397216
1473	957.45	976.599	491.982	690.5597754	1161.440225	347.8838084	347.8838084	882.7801882
1474	958.1	977.262	492.316	691.0285872	1160.971413	348.1199821	348.1199821	882.2209246
1475	958.75	977.925	492.65	691.497399	1160.502601	348.3561558	348.3561558	881.6619314
1476	959.4	978.588	492.984	691.9662108	1160.033789	348.5923294	348.5923294	881.1032091
1477	960.05	979.251	493.318	692.4350226	1159.564977	348.8285031	348.8285031	880.5447581
1478	960.7	979.914	493.652	692.9038344	1159.096166	349.0646767	349.0646767	879.986579
1479	961.35	980.577	493.986	693.3726462	1158.627354	349.3008504	349.3008504	879.4286724
1480	962	981.24	494.32	693.841458	1158.158542	349.5370241	349.5370241	878.8710386
1481	962.65	981.903	494.654	694.3102698	1157.68973	349.7731977	349.7731977	878.3136783
1482	963.3	982.566	494.988	694.7790816	1157.220918	350.0093714	350.0093714	877.7565919
1483	963.95	983.229	495.322	695.2478934	1156.752107	350.2455451	350.2455451	877.1997801
1484	964.6	983.892	495.656	695.7167052	1156.283295	350.4817187	350.4817187	876.6432432
1485	965.25	984.555	495.99	696.185517	1155.814483	350.7178924	350.7178924	876.0869819
1486	965.9	985.218	496.324	696.6543287	1155.345671	350.9540661	350.9540661	875.5309967
1487	966.55	985.881	496.658	697.1231405	1154.876859	351.1902397	351.1902397	874.975288
1488	967.2	986.544	496.992	697.5919523	1154.408048	351.4264134	351.4264134	874.4198564
1489	967.85	987.207	497.326	698.0607641	1153.939236	351.6625871	351.6625871	873.8647025
1490	968.5	987.87	497.66	698.5295759	1153.470424	351.8987607	351.8987607	873.3098268
1491	969.15	988.533	497.994	698.9983877	1153.001612	352.1349344	352.1349344	872.7552297
1492	969.8	989.196	498.328	699.4671995	1152.5328	352.3711081	352.3711081	872.2009119
1493	970.45	989.859	498.662	699.9360113	1152.063989	352.6072817	352.6072817	871.6468739
1494	971.1	990.522	498.996	700.4048231	1151.595177	352.8434554	352.8434554	871.0931161
1495	971.75	991.185	499.33	700.8736349	1151.126365	353.079629	353.079629	870.5396392
1496	972.4	991.848	499.664	701.3424467	1150.657553	353.3158027	353.3158027	869.9864437
1497	973.05	992.511	499.998	701.8112585	1150.188741	353.5519764	353.5519764	869.43353
1498	973.7	993.174	500.332	702.2800703	1149.71993	353.78815	353.78815	868.8808988
1499	974.35	993.837	500.666	702.7488821	1149.251118	354.0243237	354.0243237	868.3285506
1500	975	994.5	501	703.2176939	1148.782306	354.2604974	354.2604974	867.7764859
1501	975.65	995.163	501.334	703.6865057	1148.313494	354.496671	354.496671	867.2247053
1502	976.3	995.826	501.668	704.1553175	1147.844683	354.7328447	354.7328447	866.6732093
1503	976.95	996.489	502.002	704.6241293	1147.375871	354.9690184	354.9690184	866.1219985
1504	977.6	997.152	502.336	705.0929411	1146.907059	355.205192	355.205192	865.5710733
1505	978.25	997.815	502.67	705.5617529	1146.438247	355.4413657	355.4413657	865.0204344
1506	978.9	998.478	503.004	706.0305647	1145.969435	355.6775394	355.6775394	864.4700823
1507	979.55	999.141	503.338	706.4993765	1145.500624	355.913713	355.913713	863.9200175
1508	980.2	999.804	503.672	706.9681883	1145.031812	356.1498867	356.1498867	863.3702406
1509	980.85	1000.467	504.006	707.4370001	1144.563	356.3860604	356.3860604	862.8207522
1510	981.5	1001.13	504.34	707.9058118	1144.094188	356.622234	356.622234	862.2715527

				Vehicle A		Vehicle B		Distance (A, B)
Sample #	Time(sec)	Dist A trvl (m)	Dist B trvl (m)	x1	y1	x2	y2	
2185	1420.25	1448.655	729.79	1024.353774	827.6462259	516.0394578	516.0394578	596.2232988
2186	1420.9	1449.318	730.124	1024.822586	827.1774141	516.2756315	516.2756315	596.0536245
2187	1421.55	1449.981	730.458	1025.291398	826.7086023	516.5118052	516.5118052	595.8848267
2188	1422.2	1450.644	730.792	1025.760209	826.2397905	516.7479788	516.7479788	595.7169063
2189	1422.85	1451.307	731.126	1026.229021	825.7709787	516.9841525	516.9841525	595.5498639
2190	1423.5	1451.97	731.46	1026.697833	825.3021669	517.2203262	517.2203262	595.3837004
2191	1424.15	1452.633	731.794	1027.166645	824.8333551	517.4564998	517.4564998	595.2184163
2192	1424.8	1453.296	732.128	1027.635457	824.3645433	517.6926735	517.6926735	595.0540126
2193	1425.45	1453.959	732.462	1028.104268	823.8957315	517.9288472	517.9288472	594.8904898
2194	1426.1	1454.622	732.796	1028.57308	823.4269197	518.1650208	518.1650208	594.7278487
2195	1426.75	1455.285	733.13	1029.041892	822.9581079	518.4011945	518.4011945	594.5660901
2196	1427.4	1455.948	733.464	1029.510704	822.4892961	518.6373682	518.6373682	594.4052147
2197	1428.05	1456.611	733.798	1029.979516	822.0204843	518.8735418	518.8735418	594.2452232
2198	1428.7	1457.274	734.132	1030.448327	821.5516726	519.1097155	519.1097155	594.0861162
2199	1429.35	1457.937	734.466	1030.917139	821.0828608	519.3458892	519.3458892	593.9278946
2200	1430	1458.6	734.8	1031.385951	820.614049	519.5820628	519.5820628	593.770559
2201	1430.65	1459.263	735.134	1031.854763	820.1452372	519.8182365	519.8182365	593.6141101
2202	1431.3	1459.926	735.468	1032.323575	819.6764254	520.0544101	520.0544101	593.4585486
2203	1431.95	1460.589	735.802	1032.792386	819.2076136	520.2905838	520.2905838	593.3038752
2204	1432.6	1461.252	736.136	1033.261198	818.7388018	520.5267575	520.5267575	593.1500907
2205	1433.25	1461.915	736.47	1033.73001	818.26999	520.7629311	520.7629311	592.9971957
2206	1433.9	1462.578	736.804	1034.198822	817.8011782	520.9991048	520.9991048	592.8451908
2207	1434.55	1463.241	737.138	1034.667634	817.3323664	521.2352785	521.2352785	592.6940769
2208	1435.2	1463.904	737.472	1035.136445	816.8635546	521.4714521	521.4714521	592.5438545
2209	1435.85	1464.567	737.806	1035.605257	816.3947428	521.7076258	521.7076258	592.3945244
2210	1436.5	1465.23	738.14	1036.074069	815.925931	521.9437995	521.9437995	592.2460871
2211	1437.15	1465.893	738.474	1036.542881	815.4571192	522.1799731	522.1799731	592.0985435
2212	1437.8	1466.556	738.808	1037.011693	814.9883074	522.4161468	522.4161468	591.9518941
2213	1438.45	1467.219	739.142	1037.480504	814.5194956	522.6523205	522.6523205	591.8061396
2214	1439.1	1467.882	739.476	1037.949316	814.0506838	522.8884941	522.8884941	591.6612807
2215	1439.75	1468.545	739.81	1038.418128	813.581872	523.1246678	523.1246678	591.5173181
2216	1440.4	1469.208	740.144	1038.88694	813.1130602	523.3608415	523.3608415	591.3742523
2217	1441.05	1469.871	740.478	1039.355752	812.6442484	523.5970151	523.5970151	591.2320841
2218	1441.7	1470.534	740.812	1039.824563	812.1754366	523.8331888	523.8331888	591.0908141
2219	1442.35	1471.197	741.146	1040.293375	811.7066248	524.0693624	524.0693624	590.9504429
2220	1443	1471.86	741.48	1040.762187	811.237813	524.3055361	524.3055361	590.8109712
2221	1443.65	1472.523	741.814	1041.230999	810.7690012	524.5417098	524.5417098	590.6723997
2222	1444.3	1473.186	742.148	1041.699811	810.3001895	524.7778834	524.7778834	590.5347288
2223	1444.95	1473.849	742.482	1042.168622	809.8313777	525.0140571	525.0140571	590.3979594
2224	1445.6	1474.512	742.816	1042.637434	809.3625659	525.2502308	525.2502308	590.2620919
2225	1446.25	1475.175	743.15	1043.106246	808.8937541	525.4864044	525.4864044	590.1271271
2226	1446.9	1475.838	743.484	1043.575058	808.4249423	525.7225781	525.7225781	589.9930655
2227	1447.55	1476.501	743.818	1044.04387	807.9561305	525.9587518	525.9587518	589.8599078
2228	1448.2	1477.164	744.152	1044.512681	807.4873187	526.1949254	526.1949254	589.7276546
2229	1448.85	1477.827	744.486	1044.981493	807.0185069	526.4310991	526.4310991	589.5963064
2230	1449.5	1478.49	744.82	1045.450305	806.5496951	526.6672728	526.6672728	589.465864
2231	1450.15	1479.153	745.154	1045.919117	806.0808833	526.9034464	526.9034464	589.3363278
2232	1450.8	1479.816	745.488	1046.387929	805.6120715	527.1396201	527.1396201	589.2076985
2233	1451.45	1480.479	745.822	1046.85674	805.1432597	527.3757938	527.3757938	589.0799767
2234	1452.1	1481.142	746.156	1047.325552	804.6744479	527.6119674	527.6119674	588.953163
2235	1452.75	1481.805	746.49	1047.794364	804.2056361	527.8481411	527.8481411	588.8272579
2236	1453.4	1482.468	746.824	1048.263176	803.7368243	528.0843148	528.0843148	588.7022621
2237	1454.05	1483.131	747.158	1048.731987	803.2680125	528.3204884	528.3204884	588.5781761
2238	1454.7	1483.794	747.492	1049.200799	802.7992007	528.5566621	528.5566621	588.4550005

LIST OF REFERENCES

[1] Y. Tadjdeh. (2014, Sept.). As technology matures, new roles emerge for underwater drones. [Online]. Available: http://www.nationaldefensemagazine.org/archive/2014/September/Pages/AsTechnologyMaturesNewRolesEmergeforUnderwaterDrones.aspx

[2] *The Navy Unmanned Undersea Vehicle (UUV) Master Plan*, OPNAV N77, Department of the Navy, Washington, DC, 2004.

[3] S. P. Ouimet, "Undersea navigation of a glider UUV using an acoustic communications network," M.S. thesis, Dept. Eng. Acoustics, Naval Postgraduate School, Monterey, CA, 2004.

[4] R. Turner and E. Turner, "A two-level, protocol-based approach to controlling autonomous oceanographic sampling networks," *IEEE*, vol. 26, no. 4, pp. 654–666, Oct. 2001.

[5] J. Nicholson, "Autonomous optimal rendezvous of underwater vehicles," Ph.D. dissertation, Dept. Eng., Naval Postgraduate School, Monterey, CA, 2004.

[6] W. Marr, "Acoustic based tactical control of underwater vehicles," Ph.D. dissertation, Dept. Eng., Naval Postgraduate School, Monterey, CA, 2003.

[7] G. Burrowes and J. Khan. (2011). *Short-range underwater acoustic communications networks* [Ebrary version]. [Online]. Available: http://www.intechopen.com/books/autonomous-underwater-vehicles

[8] T. Curtin et al., "Autonomous oceanographic sampling networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.

[9] D. Jiang, et al. "Coordinated control of multiple autonomous underwater vehicle system," presented at the 8th World Congress on Intelligent Control and Automation, Heilongjiang Province, China, 2010.

[10] S.C. Botelho and R. Alami, "M+ : A scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, 1999, vol. 2, pp. 1234–1239.

[11] P. Newman, *MOOS - Mission Orientated Operating Suite*, ver. 2.1, Dept. Ocean Eng. MIT, Boston, MA, 2006, pp. 1–15.

[12] E. Fiorelli et al., "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE*, vol. 31, no. 4, pp. 935–948, Oct. 2006.

[13] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Int. J. Robotics Res*, vol. 5, no. 1, pp. 90–98, spring 1986.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California